Optimal Shipping Schedules for Collaborating Blood Banks

Peter Mugaba Noertoft Zahir Ahmed

***Abstract*—The supply and demand of blood at blood banks is inherently uncertain in time. The consequences of failing to meet blood demand are disastrous and potentially fatal. In this project, we demonstrate that blood banks in a network can collaborate to reduce the amount of blood demand they miss, and provide a model predictive control inspired strategy that generates blood shipping schedules between the banks. Our proposed method has two steps, performed daily– forecasting supply and demand of blood at each blood bank, then forming an optimal control problem to find shipping schedules. We also demonstrate a forecasting method that learns joint structure in supply and demand across blood banks, performing better than baseline models that consider blood banks individually.**

I. Introduction

Blood banks are medical facilities that collect, process, and store blood. They play a critical role in healthcare, supplying the blood necessary for lifesaving procedures. One key challenge in their operation is inherit uncertainty in the demand and supply of blood, which can result in disastrous consequences. We propose that this problem can be mitigated by *collaboration* in a network of blood banks shipping blood between each other: the crucial question being how to set effective shipping schedules.

Learning-based models for health services have appeared recently in the literature and shown promise– in particular, to study uncertainty in the reliability of blood data via K-nearest neighbor regression [5] and forecast data for a single blood bank through extreme gradient boosting [8]. Broadening these methods, we devise a two-faceted approach– time-series methods to forecast supply and demand of blood at blood banks on a daily schedule, then designing optimal blood shipping schedules between the banks.

We proceed as follows. First, we introduce our forecasting methods and standardize notation. Next, we outline a model predictive control strategy for dynamically finding shipping schedules between banks [10]. Finally, we test our method on a toy example, and analyze results.

II. Forecasting Methodology

Supply and demand is predicted through five time-series forecasting methods. These are broadly split into two groups– non-joint models that do not account for the correlated structure of the supply-demand data in the network, and joint models that do.

We begin by fixing some terminology and notation. Define the network graph $G = (V, E)$, with each node $i \in V$ corresponding with one of the N blood banks and each edge $e_{ij} \in E$ corresponding with the distance between blood banks i and j, weighted by some cost w_{ij} thusly. For any time t and blood bank i, each forecaster produces vector $\hat{\mathcal{S}}_t^i, \hat{\mathcal{D}}_t^i \in \mathbb{R}^f$, where f is the forecast length into the future. We analogously define $\mathcal{S}_t^i, \mathcal{D}_t^i \in \mathbb{R}$ as the true values. Henceforth, we assume that at time step t, each blood bank already has access to the information of $\mathcal{S}_{t'}^i$ and $\mathcal{D}_{t'}^i$ for $t' \leq t$.

A. Non-Joint Baseline Forecasters

We begin with our non-joint forecaster examples– the zero-order hold last and historical mean– which we used as baselines to compare with the more advanced methods. In the zero-order hold forecaster, we define

$$\begin{cases} \hat{\mathcal{S}}_t^i = (\mathcal{S}_t^i, ..., \mathcal{S}_t^i), \\ \hat{\mathcal{D}}_t^i = (\mathcal{S}_t^i, ..., \mathcal{S}_t^i), \end{cases}$$

i.e., forecasting that the values will remain fixed at the last measurement. For the historical mean forecaster, fix an integer $h \geq 1$ representing the number of timesteps to look back; define $\overline{\mathcal{S}}_t^i = \frac{1}{h}(\mathcal{S}_t^i + \mathcal{S}_{t-1}^i + \cdots + \mathcal{S}_{t-h+1}^i)$ and $\overline{\mathcal{D}}_t^i$ analogously. Then

$$\begin{cases} \hat{\mathcal{S}}_t^i = (\overline{\mathcal{S}}_t^i, ..., \overline{\mathcal{S}}_t^i), \\ \hat{\mathcal{D}}_t^i = (\overline{\mathcal{S}}_t^i, ..., \overline{\mathcal{S}}_t^i), \end{cases}$$

which fixes the forecast to the mean of the last h days.

B. Auto-Regressive Forecaster (joint)

For our joint models, we will be henceforth fixing an integer history hyper-parameter $h \geq 1$ that our models take as input, mirroring the situation of the mean forecaster. To compute a forecast $\hat{\mathcal{S}}_t^i, \hat{\mathcal{D}}_t^i$, we construct a model, and train it on a subset of the known data $\mathcal{S}_{t'}^i, \mathcal{D}_{t'}^i$ for $t' < t - f$ (the *training data*).

In terms of a joint baseline, we implement an auto-regressive forecaster. Fixing any time t, take $\hat{\mathcal{Y}}_t \in \mathbb{R}^{2fN}$ as a concatenation of all the $\hat{\mathcal{S}}_t^i$'s and $\hat{\mathcal{D}}_t^i$'s. Similarly take $\mathcal{X}_t \in \mathbb{R}^{2N}$ as the concatenation of all the $\mathcal{S}_t^i$'s and $\mathcal{D}_t^i$'s. We assume that $\hat{\mathcal{Y}}_t$ depends linearly on the previous h values of $\mathcal{X}_t$, i.e.,

$$\hat{\mathcal{Y}}_t = \alpha + \beta_1 \mathcal{X}_{t-h+1} + \cdots + \beta_h \mathcal{X}_t$$

for some choice of coefficient matrices $\beta_\lambda \in \mathbb{R}^{2fN \times 2N}$ and intercept vector $\alpha \in \mathbb{R}^{2fN}$. We fit these model parameters on the train data, minimizing least square error.

Peter Mugaba Noertoft is submitting this project to CS 229, Fall 2024.
Zahir Ahmed is submitting this project to CS 238, Fall 2024.

C. Bayesian Vector Autoregression (BVAR)

The setup for BVAR is similar to the auto-regressive forecaster detailed above. Taking $\mathcal{X}_t$ as before, we assume $\mathcal{X}_{t+1}$ is approximately linear with respect to the previous h values of $\mathcal{X}_t$, i.e.,

$$\mathcal{X}_{t+1} = \epsilon_t + \alpha + \beta_1 \mathcal{X}_{t-h+1} + \cdots + \beta_h \mathcal{X}_t.$$

As before, we have intercept $\alpha \in \mathbb{R}^{2N}$ and linear coefficient matrices $\beta_\lambda \in \mathbb{R}^{2N \times 2N}$. We have also added a time-dependent error term $\epsilon_t \in \mathbb{R}^{2N}$, where we assume each entry $\epsilon_t^i \sim \mathcal{N}(0, \sigma_i^{\text{noise}})$ for some choice of variances σ_i^{noise}. The idea behind BVAR, observed through the economic forecasting work of [7] and [9], is to encapsulate the uncertainty of these parameters by treating them as random variables, where in training we perform Bayesian updates initialized from a manually-set prior distribution for each parameter. In our situation, we assume each entry b_{ij}^λ of B^λ is distributed $\sim \mathcal{N}(\mu_{ij}^\lambda, \sigma_{ij}^\lambda)$; similarly that each entry a_i of the intercept α is distributed $\sim \mathcal{N}(\mu_i^{\text{int}}, \sigma_i^{\text{int}})$ and each $\sigma_i^{\text{noise}} \sim |\mathcal{N}|(\sigma_i^{\text{noise var}})$. Using these model definitions, we implement and train our model through PyMc [1] to obtain posterior parameter distributions. In turn, this yields a distribution forecasts by iteratively computing $\mathcal{X}_{t+1}, ..., \mathcal{X}_{t+f}$, as illustrated below.

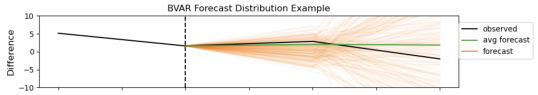


Fig. 1. Forecasting via BVAR

D. Gradient Boosting

The set up for the gradient boosting forecaster is the same as that of the auto-regressive forecaster. However, rather than assuming that $\hat{\mathcal{Y}}_t$ is a linear function of the previous h values of $\mathcal{X}_t$, we build an ensemble of regression decision trees that combine to learn a joint non-linear mapping from past blood supply and demand values at each blood bank to future blood supply and demand values at each blood bank. The regression decision trees are fit sequentially, meaning that each successive tree attempts to address the weaknesses in the existing ensemble. We leverage XGBoost to implement regularized gradient boosting [4].

III. Model Predictive Control (MPC)

Given forecast data $S \in \mathbb{R}^{N \times f}, D \in \mathbb{R}^{N \times f}$ at a fixed time t_0 of f steps into the future, we wish to produce shipment amounts Q_t^{ij} from bank j to bank i at times $t = 0, ..., f-1$ (relative to t_0), where shipments are standardized to arrive the day after they are shipped. We formulate this as a convex optimization problem. In particular, we define $B_t^i \in \mathbb{R}$ as the amount of blood in the bank i at time t, and $M_t^i \in \mathbb{R}$ as an exogenous "loan" variable, used as an infinite source and sink blood bank. Define Q, B, and M as the tensors associated with the respective quantities defined above. We further take exogenous parameters as follows– a discount factor $\gamma \in (0, 1)$ to incentivize meeting short-term demand, the weight matrix $W = (w_{ij})$ associated with the network structure, a maximum blood storage capacity $b_{\max} \in \mathbb{R}^N$, a maximum transportation amount $Q_{\max} \in \mathbb{R}$, an initial blood storage distribution $B_{\text{init}} \in \mathbb{R}^N$, and a scaling parameter $\alpha \in \mathbb{R}_{\geq 0}$. Putting everything together, we formulate

$$\begin{array}{ll} \text{minimize} & \sum_i ||\Gamma M^i||_1 + \alpha \sum_t ||W \odot Q_t||_2 \\ \text{subject to} & B_0 = B_{\text{init}} \\ & 0 \preceq B_t \preceq b_{\max}, \forall t \\ & B_{t+1}^i = B_t^i + S_t^i - D_t^i + \sum_j Q_t^{ij} + M_t^i, \forall i, t \\ & Q^{ij} = -Q^{ji}, \forall i, j \\ & \sum_j Q_t^{ji} \leq B_t^i, \forall i, t \\ & ||Q||_\infty \leq Q_{\max}, \end{array}$$

where Q, B, and M are the optimization variables, $\Gamma = \text{diag}(1, \gamma, \cdots, \gamma^{f-1})$, and $\odot$ is the entry-wise matrix product. The objective function penalizes using the "loan" bank, as well as making shipments far away. The constraints can loosely be categorized as follows– initial and boundary constraints for B, storage dynamics, total blood is conserved, a bank cannot ship more blood than it has in stock, and shipments cannot exceed a maximum daily limit. We leverage CVXPY to solve this problem [2], [6].

Combining everything, given a forecaster, we obtain an MPC loop as follows. At time t, obtain forecasts $\hat{\mathcal{S}}_t, \hat{\mathcal{D}}_t \in \mathbb{R}^{N \times f}$. Using the optimization problem above, we can extract the forecasted optimal shipment $Q_t^* \in \mathbb{R}^{N \times N}$ and "loan" $M_t^* \in \mathbb{R}^N$ at our time (note: crucially, here Q_t^* and M_t^* are *not* the matrices Q_t and M_t from the optimization, but rather the action and loan prescribed at time 0 by the optimizer). We update the total amount of blood in the banks $B_t \in \mathbb{R}^N$ through

$$B_{t+1}^i = B_t^i + \mathcal{S}_{t+1}^i - \mathcal{D}_{t+1}^i + \sum_j (Q^*)_t^{ij} + (M^*)_t^i$$

for all i. Clip each B_{t+1}^i to lie between 0 and $b_{\max}^i$; further define a clip vector $C_t \in \mathbb{R}^N$ that equals the magnitude of the clip if $B_{t+1}^i < 0$ and equals zero otherwise. Passing B_{t+1}^i as the new B_{init} for our optimizer, we can then iterate this process indefinitely.

Using this data, we can compute a realized cost of the forecaster mimicking the form of the loan M_t in the optimizer. In particular, choosing another real parameter β, we take the realized cost $\mathcal{E}_T$ as

$$\mathcal{E}_T = \frac{1}{T} \sum_{t=0}^{T} \Big(||M_t^*||_1 + \alpha ||W \odot Q_t^*||_2 + \beta ||C_t||_1 \Big),$$

where T is the number of days scheduled for, so as to appropriately penalize taking loans and dumping blood, sending shipments far, and missing blood demand.

IV. Experiment

To test our methodology, we consider a toy example of $N = 4$ California blood banks sampled from [3], with weights

$$W = (w_{ij}) = \begin{pmatrix} 0 & 35.4 & 403 & 403 \\ 35.4 & 0 & 370 & 370 \\ 403 & 370 & 0 & 1.4 \\ 403 & 370 & 1.4 & 0 \end{pmatrix}$$

the distance in miles between blood bank i and j. A visualization of this network is depicted below.

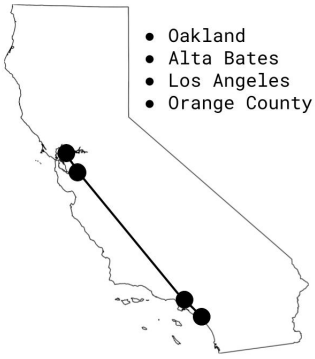


Fig. 2. Blood Bank Network

A. Data Generation

The pertinent issue for data acquisition is the fundamental lack of publicly available blood supply and demand data. Indeed, our only sample stems from [5], giving us monthly supply and demand data from 2013 through 2020 in the Tema General Hospital of Ghana. We thus have two problems to solve– interpolate the source data to a daily scale; and generate synthetic, but realistic, daily data for our toy problem.

For the first, we perform a cubic interpolation to smoothly connect the monthly data points, yielding us the supply and demand for a single blood bank over T daily time steps. A linear interpolation was also attempted, but yielded undesirable jagged piecewise behavior.

For the second, we perform a sequence of procedures. We begin by scaling our given supply and demand data by a population ratio, so we can obtain a rough estimate on the scale of supply and demand in the area. To generate more interesting data where cumulative supply does not always exceed the cumulative demand, we further randomly swap the supply and demand of some of the blood banks in the network. Finally, we incorporate zero-mean Gaussian noise into the supply and demand data. In particular, for two times t, t' and blood banks i, i', we define the *covariance entry*

$$s_{iti't'} = \alpha e^{-w_{ii'}/\beta} + (1-\alpha)e^{-|t-t'|/\gamma}$$

for parameters $\alpha = 0.9, \beta = 30, \gamma = 10$ to control scaling and decay for the spatial and temporal relationships respectively. We then reshape to a $NT \times NT$ matrix Σ, and update $\Sigma \leftarrow \sqrt{\Sigma^T \Sigma}$ to ensure symmetry and positive-definiteness.

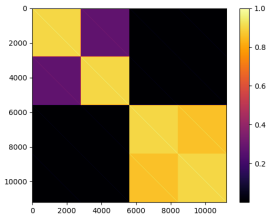
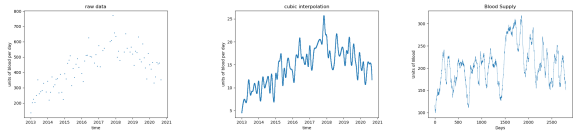


Fig. 3. Covariance Structure of Σ

Each tile in the color plot indicates the magnitude of correlation between blood bank i and j– particularly in regard to spatial proximity– with the diagonal bands indicating the temporal relationship between blood banks.

Following this, we take two NT-vector samples $\sim \mathcal{N}(0, \Sigma)$ as our additive correlated noise. The entire data generation process for the supply of a particular blood bank is depicted in Figure 4.



(a) Raw (b) Interpolate (c) Noise

Fig. 4. Data Generation for Blood Banks

B. Experimental Results

In this section, we describe the experimental results applying the forecasting methods and the model predictive control strategy to the aforementioned data. We fix the network parameters $b_{\text{max}} = (2000, 2000, 2000, 2000)$, $b_{\text{init}} = (0, 0, 0, 0)$, $Q_{\text{max}} = 100$, $\gamma = 0.999$, $\alpha = 10^{-3}$, and $\beta = 1$.

We split the data into three groups: train, validation, and test, maintaining chronological order to avoid information leakage. The train group consists of the first five years of data and is used to train all learning models. The validation group consists of the next two years of data, and is used to selected the hyper-parameters with best performance on unseen data. The test group consists of the final year of data and remains unseen by all models until they are both trained and have fixed hyper-parameters. We report forecaster root-mean-square error (RMSE) and end-to-end realized cost $\mathcal{E}$ of the forecast-decision strategies on test set for one week forecasts ($f = 7$).

During our exploratory data analysis, we found strong seasonal trends in both the supply and demand of blood at all blood banks, so the historical mean forecaster (non-joint), the auto-regressive forecaster, and the gradient boosting forecaster all performed better with features consisting of changes in supply and demand of blood at each blood bank (i.e., first difference signals). We henceforth train our models on first difference signals, reverting back to total signals for calculation of the appropriate RMSE.

1) Zero-Order Hold Forecaster: The Zero-Order Hold forecaster needs no training and has no hyper-parameters to select and is therefore run directly on the test data.

2) Historical Mean Difference Forecaster: Computing the validation RMSE for the range $h \in [2, 100]$, we choose $h = 12$. (note: validation RMSE vs h plot shown in appendix).

3) Auto-Regressive Forecaster: Computing the validation RMSE for the hyper-parameter range $h \in [2, 100]$, we choose $h = 11$ (note: validation RMSE vs h plot shown in appendix).

4) Baysian Vector Auto-Regressive Forecaster: Computing the validation RMSE for the hyper-parameter range $h \in [2, 15]$, we choose $h = 12$. All priors are set $\sim \mathcal{N}(0, 10)$ to indicate a fairly large parameter uncertainty.

5) Gradient Boosting Forecaster: The gradient boosting forecaster has several hyper-parameters that need to be selected. We performed a grid search of hyper-parameters $h \in [2, 40]$, `maximum tree depth` $\in [1, 8]$, `learning rate` $\in [0.001, 1]$, and `minimum child weight` $\in [0.00, 10]$. The selected parameters based on lowest validation error are $h = 13$, `maximum tree depth` $= 1$, `learning rate` $= 0.5$, and `minimum child weight`$= 0$.

On our test data set, we obtain the following RMSE values for each forecaster:

Forecaster	Test RMSE
Zero-Order Hold	6.48
Historical Mean	6.46
Auto-Regressive	4.60
BVAR	5.06
Gradient Boosting	4.57

TABLE I
ROOT MEAN SQUARE ERROR ON TEST SET

Observe that the joint learning-based models perform better than the non-joint, non-learning baselines in this metric.

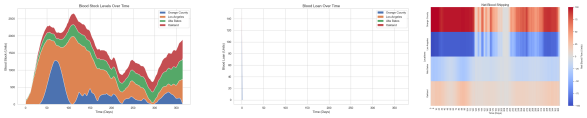
There is a another, more intrinsic, way of measuring forecaster efficacy– the realized cost $\mathcal{E}_{365}$ of each forecaster over the test year. We provide the $\mathcal{E}_{365}$ costs for each forecasting method below, further including forecasters with perfect future information as points of comparison.

Model	Cost $\mathcal{E}_{365}$
Perfect Knowledge (no shipment)	154.45
Perfect Knowledge (with shipment)	2.92
Zero-Order Hold Forecaster	3.69
Historical Mean Forecaster	3.65
Auto-Regressive Forecaster	3.56
BVAR Forecaster	4.04
Gradient Boosting Forecaster	3.76

TABLE II
MPC LOOP REALIZED COST

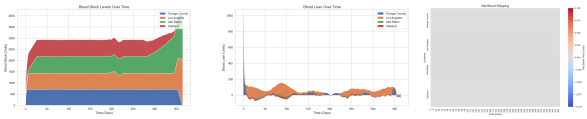
In particular, observe that though the auto-regressor continues to perform well, the BVAR and Gradient Boosting forecasters fall short as compared to the non-joint baselines. The perfect knowledge forecaster with shipment expectedly performs the best; on the other hand, the perfect knowledge forecaster without shipment performs by far the worst as compared to all forecasting methods. This lends credence to the power of collaboration between blood banks and the effectiveness of our forecasting-to-network-shipment methodology.

Through the MPC loop, we further obtain representations of the total blood storage B, the loaning M, and the actions Q occurring each day. These are displayed in Figures 7,8,9, 10, and 11, along with both the perfect information cases in Figures 5. and 6 To represent the total blood stock in the blood banks on any given day in the test data, we use a stack plot, with the vertical axis denoting the total stock in storage and the horizontal axis denoting the elapsed time. To represent M, we use another stack plot with regards to the four blood banks, with the vertical axis depicting the loan/dump at time t. Finally, to represent Q, we use a heat map where the color at a point (t, i) indicates the total magnitude of blood shipped to bank i at time t (note: possibly negative when i is a net exporter).



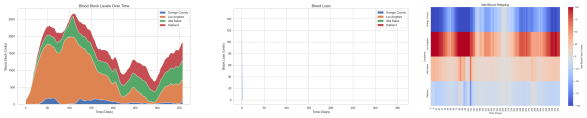
(a) B (b) M (c) Q

Fig. 5. Plots of Variables for Perfect Information (with shipment)



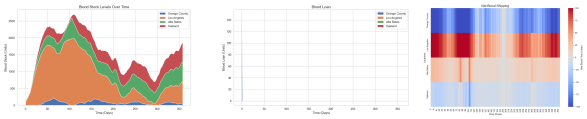
(a) B (b) M (c) Q

Fig. 6. Plots of Variables for Perfect Information (without shipment)



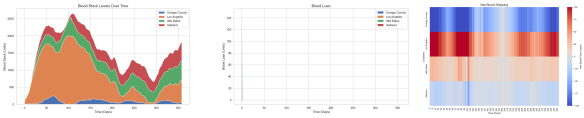
(a) B (b) M (c) Q

Fig. 7. Plots of Variables for Zero-Order Hold Forecaster



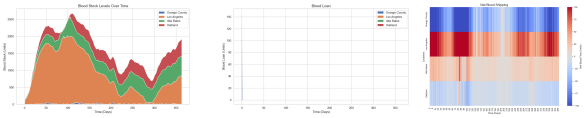
(a) B (b) M (c) Q

Fig. 8. Plots of Variables for Historical Mean Forecaster



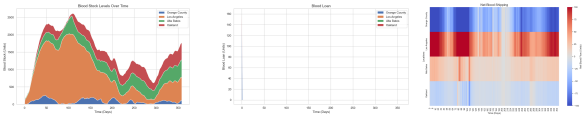
(a) B (b) M (c) Q

Fig. 9. Plots of Variables for Auto-Regressive Forecaster



(a) B (b) M (c) Q

Fig. 10. Plots of Variables for BVAR Forecaster



(a) B (b) M (c) Q

Fig. 11. Plots of Variables for Gradient Boosting Forecaster

V. Conclusion & Future Work

We have demonstrated an effective methodology of addressing uncertainty in blood supply and demand over a network of blood banks. Relative to perfect knowledge with no shipment, our forecasting-and-MPC-based blood shipment strategy remains competitive with the unrealistic perfect knowledge model with shipment.

On the pure forecasting task, all the joint forecasters outperform the non-joint baseline models on unseen test data. However, on the end-to-end performance task, the BVAR and Gradient Boosting forecasters lag behind the non-joint baseline models, while the auto-regressive model maintains its dominance. We suspect that the BVAR and gradient boosting forecasters are either over-fitting to seen data, or have a bias to under-forecasting (as opposed to over-forecasting), which is heavily penalized by the end-to-end test– this warrants further exploration.

Another avenue for future exploration is developing more robust formulations of MPC for optimal shipping schedules. For example, probabilistic forecasters that predict distributions of future outcomes rather than single points can help capture the uncertainty based risk associated with a given shipping schedule. Modifications to the MPC objective like jointly optimizing for multiple quantiles or samples of the forecast distribution could then yield shipping schedules that are more robust to outlier scenarios.

Alternatively, the problem could be extended through loosening the restriction that each blood bank can observe the supply and demand of all other blood banks– instead, we can consider the situation of partial (local) observability by each blood bank, but such that the collection of all observations determine the state. This setup can be formulates as a Decentralized Markov Decision Process, which lends itself to a vast array of new methods to extract optimal policies.

VI. Contributions, Ethics, and Reproducibility

PMN and ZA contributed equally to all parts of this project, including idea formation, data generation, forecaster and optimal control implementation, and paper writeup.

This project provides an exploratory analysis for a toy example. The primary purpose is education. The authors do not advise to implement any of the contained methods for a real blood bank network without comprehensive additional testing.

For reproducibility of results, all code is available at `github.com/zahmed2003/BloodFlow`.

References

[1] Abril-Pla O, Andreani V, Carroll C, Dong L, Fonnesbeck CJ, Kochurov M, Kumar R, Lao J, Luhmann CC, Martin OA, Osthege M, Vieira R, Wiecki T, Zinkov R. 2023. PyMC: a modern, and comprehensive probabilistic programming framework in Python. *PeerJ Computer Science* 9:e1516 https://doi.org/10.7717/peerj-cs.1516

[2] Agrawal, A., Verschueren, R., Diamond, S., & Boyd, S. (2018). A rewriting system for convex optimization problems. Journal of Control and Decision, 5(1), 42-60.

[3] "Blood Establishment Registration Database", U.S. Food & Drug Administration

[4] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

[5] Clement Tswusami and Juliet Tswusami (2022) "Machine learning algorithms for forecasting and backcasting blood demand data with missing values and outliers: A study of Tema General Hospital of Ghana", International Journal of Forecasting

[6] Diamond, S., & Boyd, S. (2016). CVXPY: A Python-embedded modeling language for convex optimization. Journal of Machine Learning Research, 17(83), 1-5.

[7] Doan, T., Litterman, R. B., & Sims, C. A. (1984). Forecasting and Conditional Projection Using Realistic Prior Distributions. *Econometric Reviews*, 3, 1–100.

[8] Hi Jeong Kwon et al. (2024) "Development of blood demand prediction model using artificial intelligence based on national public big data", Sage Journals

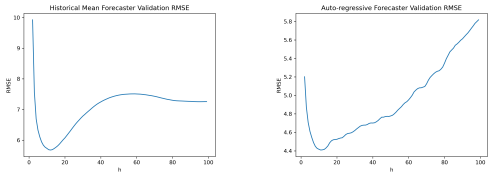
[9] Litterman, R. B. (1986). Forecasting with Bayesian Vector Autoregressions: Five Years of Experience. *Journal of Business & Economic Statistics*, 4, 25–38.

[10] Schwenzer, M., Ay, M., Bergs, T. *et al*. Review on model predictive control: an engineering perspective. *Int J Adv Manuf Technol* 117, 1327–1349 (2021).

VII. Appendix

A. RMSE validation plots

We show the validation RMSE plots used to select the hyper-parameter history length h for the historical mean and auto-regressive forecasters.



(a) Historical Mean (b) Auto-regressor

Fig. 12. Plots of h-hyperparameter Fitting

B. Auto-regressive model parameters

We show the stacked auto-regressive model coefficients $\beta \in \mathbb{R}^{2Nf \times 2Nh}$ for $h = 11$. The coefficients exhibit both time and spatial structure.

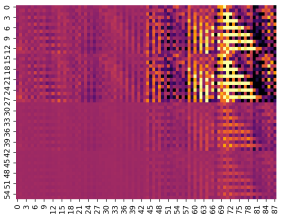


Fig. 13. Auto-regressive model coefficients