Sample Paper 1 - Human Knowledge Guided Advanced Analytics

A Robust and Trustable Approach to Incorporate Medical Domain Knowledge in Machine Learning Models for Diabetic Retinopathy Screening Using Routine Lab Results

Enrico Laoh [1] *; **Oday Bani Ahmad** [2]**; Tieming Liu** [3] *

[1] elaoh@okstate.edu, School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK 74078

[2] obaniah@okstate.edu, School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK 74078

[3] tieming.liu@okstate.edu, School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK 74078

* corresponding author

Keywords: Healthcare Analytics, Diabetic Retinopathy Prediction, Elaborative Learning, Trustable AI, PubMed

Abstract

Most AI models for disease prediction are difficult to integrate into clinical practice due to their complexity and lack of interpretability. There is a critical need for accurate, self-interpretable models that align with clinical reasoning and minimize cognitive burden for physicians. This study aims to develop a robust and trustable disease prediction model that overcomes this hurdle. To counteract the potential bias introduced by a limited number of participating physicians, we design a new bibliometric statistic metric to extract medical domain knowledge through PubMed at the National Library of Medicine and identify lab variables that are widely accepted as risk factors for the disease. Then, a novel elaborative learning approach is proposed to account for human cognitive limitations in processing information to ensure the usability of the model by physicians. The disease of focus in this study is diabetic retinopathy, a potentially sight-threatening complication of diabetes. The study demonstrates a 96.65% accuracy and a 96.60% F1-score in predicting diabetic retinopathy. This is accomplished using only eight features that align with the medical domain knowledge, indicating the model's efficacy in identifying the disease. The model's good performance underscores its potential as a screening tool for primary care physicians, specifically for conditions like diabetic retinopathy, where early detection is critical. The eight variables also comply with the human cognitive limit to digest information simultaneously. This research offers a promising direction for using AI in healthcare, particularly in disease detection. It presents a scalable approach for implementing robust and trustable AI for other diseases.

1. Introduction

Although there has been an increase in studies using machine learning (ML) for disease prediction, their implementation in routine clinical practice is still limited. Most AI-based tools that have received approval from the U.S. Food and Drug Administration (FDA) are primarily focused on radiology and image-based diagnostics, with little adoption in non-imaging primary care environments. A key barrier to clinical adoption is the lack of trust, particularly from physicians, due to the high-stakes nature of medical decision-making. Clinicians are expected to make critical decisions and need models that provide accurate predictions along with clear and understandable reasoning. While recent advances in explainable AI (XAI), such as attention maps and feature attribution tools, have improved post-hoc explanation capabilities, they often fail to bridge the gap in clinician-facing interpretability, that is, the ability for physicians to directly validate and comprehend model decisions without relying on external interpretation tools. In this study, we emphasize self-interpretable models, where the logic and decision pathway are inherently understandable.

We adopted the model to be self-interpretable to support transparency and alignment with clinical reasoning. The model relies on a small set of routinely collected, medically validated features and leverages domain knowledge through a structured selection process. This design aims to reduce cognitive burden and improve the model’s potential usability in real-world settings. Importantly, we demonstrate that this simplified, interpretable approach achieves strong predictive performance, showing that high accuracy and interpretability need not be mutually exclusive. Incorporating medical domain knowledge into ML models for healthcare presents a major challenge. Physicians who will ultimately employ these models are often occupied with their daily responsibilities, leaving them with limited bandwidth to actively engage in research efforts or provide feedback on the explainability of the AI models. The small number of participating physicians is insufficient to adequately represent the entire medical domain.

A comprehensive strategy will be employed to achieve an accountable disease prediction model. First, we design a new bibliometric statistic metric to extract medical domain knowledge through PubMed at National Library of Medicine and identify lab variables that are widely accepted as risk factors for the disease. This counteracts the potential bias introduced by a limited number of participating physicians. This new metric offers a comprehensive and in-depth assessment of each variable by incorporating various aspects, including the term frequency in the documents, the number of citations received by the document, citation practices in the field, and the age of the document. Second, we designed a novel ML framework inspired by elaborative learning, a cognitive learning approach to improve understanding (Levin, 1988). This method encourages learners (the machine learning model) to link new data with prior knowledge, resulting in a more in-depth comprehension of the historical information.

2. Literature Review

The focus of this paper is diabetic retinopathy (DR), a complication of diabetes that affects small blood vessels. It is the most common cause of vision loss among diabetic patients and is the leading cause of blindness in American adults (Flaxel et al., 2020). In 2020, 9.5 million Americans were affected by diabetic retinopathy (DR), and this number is anticipated to rise to 16 million by 2050, with 3.4 million individuals at risk of blindness (National Diabetes Statistics Report, 2020). Almost all patients with type 1 diabetes and over 60% of patients with type 2 diabetes develop diabetic retinopathy (DR) after 20 years (American Diabetes Association, 2023).

In spite of the widespread occurrence of diabetic retinopathy (DR), adherence to the advised yearly eye exams is disturbingly low at approximately 43% (Fisher et al., 2016). Consequently, about 25% of DR patients and 19% (Kovarik et al., 2016) with potentially severe DR remains unidentified. Many individuals with diabetes do not pursue necessary medical evaluations since DR often shows no symptoms in its early phases, even though significant pathology may already be present. Diabetic retinopathy (DR) is treatable, but vision loss cannot be reversed. Early detection of DR can help reduce the need for costly treatments, such as vitrectomy surgery, for many patients in advanced stages of the disease. Additionally, the availability of ophthalmologists and necessary diagnostic equipment is mainly concentrated in urban centers, further limiting access to essential eye care for diabetic individuals in rural areas.

The rising prevalence of diabetes, combined with challenges in accessing eye exams, underscores the urgent need for affordable and accessible tools for diabetic retinopathy (DR) detection that do not rely on specialized equipment. Our goal is to develop an artificial intelligence (AI) tool that utilizes comorbidity data and routine laboratory results from the primary care visits of diabetic patients for screening, detection, and prevention of DR. This tool will empower primary care physicians (PCPs) to evaluate the risk of DR in their patients, recommend necessary eye exams, and establish tailored screening intervals for those at risk. Consequently, patients with asymptomatic DR can be treated effectively in the early stages, preventing vision loss. Our approach is cost-effective and widely available because the data needed already exist for most diabetic patients.

Several studies have shown that early diagnosis is crucial to improving DR patient outcomes (Mersha et al., 2022; Mrugacz et al., 2021; Ribeiro et al., 2016; Ting et al., 2017; Yau et al., 2012). For example, Yau et al. revealed that early management greatly lowers the probability of eyesight loss in diabetes individuals (Yau et al., 2012). The difficulty is in the plethora of variables that might impact the start and course of diabetic retinopathy, such as blood sugar levels, blood pressure, lipid profiles, and numerous demographic factors. Machine learning has emerged as a powerful method for unlocking the predictive power contained within these many factors. Ting et al. showed the power of machine learning methods, such as deep learning neural networks, in harnessing these characteristics to predict diabetic retinopathy accurately (Ting et al., 2017). This demonstrates machine learning's transformational potential in changing the early diagnosis of diabetic retinopathy, a breakthrough with far-reaching consequences for public health and patient care.

The machine learning models that can produce both high-quality and intelligible predictions are demanded by healthcare practitioners (Arbelaez Ossa et al., 2022; Caruana et al., 2015; Lee & Yoon, 2021; Lundberg et al., 2018; Rajkomar et al., 2018; Shailaja et al., 2018; Westerlund et al., 2021). Rajkomar et al. have shown the benefits of interpretable models in clinical settings, notably when dealing with EHRs and patient outcome prediction (Rajkomar et al., 2018). Caruana et al. emphasized the importance of model interpretability in medical risk prediction, highlighting the necessity to balance complexity and transparency (Caruana et al., 2015). Shailaja et al. have examined the influence of explainable ML models in radiology, where interpretability is critical in diagnosing medical imaging data (Shailaja et al., 2018). This research highlights the need to provide healthcare practitioners with models that provide detailed explanations for their predictions.

There is an increasing amount of research on the model explainability in the healthcare domain recently. Markus, Kors, and Rijnbeek stressed the importance of designing AI systems with explainability at their core to foster trust among clinicians, and they proposed a comprehensive framework for selecting between different explainable AI methodologies (Markus et al., 2021). Similarly, Wang and Yin found that the effectiveness of explanations in AI-assisted decisions greatly depends on the individual's domain expertise, with certain types of explanations, such as feature contribution, enhancing understanding, and trust in more knowledgeable users (X. Wang & Yin, 2021).

Amann et al. highlight the nuanced debate around the necessity of explainability in clinical decision support systems, suggesting that its value is contingent upon various factors, including technical feasibility and the specific context of use (Amann et al., 2022).

However, making the model explainable does not come without a cost. Holzinger et al. addressed the "black box" problem, emphasizing the difficulties in comprehending the inner workings of sophisticated ML algorithms even though they outperform others (Holzinger et al., 2019). Abedin presents strategies to sacrifice the model performance to improve its trustworthiness (Abedin, 2021).

In short, the focus of most research is either increasing the model performance or maintaining the interpretability. Very few of them aim for the two objectives simultaneously, which is needed to make our prediction model adaptable by the end user. Our study model framework prioritizes interpretable output while maintaining prediction quality, encouraging trust as needed for incorporation into clinical decision-making processes.

Furthermore, the human ability to comprehend information presents another challenge in establishing confidence in machine learning (ML) models, particularly in the context of complicated healthcare predictions. Miller's landmark study from 1956 shed light on human cognitive limitations, indicating that individuals can successfully organize and grasp just around seven variables at once, give or take two, depending on their cognitive ability (Miller, 1956). This fundamental insight highlights the difficulty in communicating complicated ML model results to healthcare practitioners in an understandable way. Aside from these cognitive limits, validating model predictions against known medical knowledge is critical for creating trust. Wang et al. have shown the relevance of correlating ML predictions with existing medical expertise, as such validation not only improves the model's credibility but also allows clinicians to interpret and accept the predictions. This aspect has also not been implemented yet in any research on making a disease prediction model (F. Wang et al., 2019). In this study, we aim to address the gap between human cognitive limitations and the need for validation from existing domain knowledge. We will consider both of these factors as essential elements in assessing the performance and reliability of machine learning models for disease prediction.

3. Methods and Materials

3.1. The data set.

We used data from the Cerner Health Facts Data Warehouse. The study cohort includes diabetic patients aged 18 to 65. Patients with diabetes were identified using ICD-9-CM and ICD-10-CM diagnosis codes (250.x, E10.x, and E11.x). Diabetic retinopathy (DR) cases were flagged based on diagnosis codes 362.0x, E10.31x–E10.35x, or E11.31x–E11.35x; all others were classified as non-DR (control) patients. Data preparation included standard preprocessing steps such as cleaning, merging, and imputing. Lab patient data with high levels of missing data or deemed irrelevant to DR prediction were excluded, resulting in a final dataset of 97,786 patients.

Gender distribution in this final cohort was 57.06% female, 42.91% male, and 0.03% unknown. Racial demographics were 62.66% Caucasian, 19.71% African American, 7.85% Other, 3.29% Asian, 3.08% Biracial, 1.49% Hispanic, 1.07% Native American, 0.84% Middle Eastern Indian, and 0.01% Pacific Islander. These distributions reflect the heterogeneity of patients across diverse clinical sites nationwide, enhancing the model's potential generalizability in real-world primary care environments.

3.2. Elaborative Learning Framework.

To build the physician's trust in the machine learning model, we need to ensure that it is interpretable and 'easy' for humans to digest. For this purpose, a novel elaborative learning framework is developed.

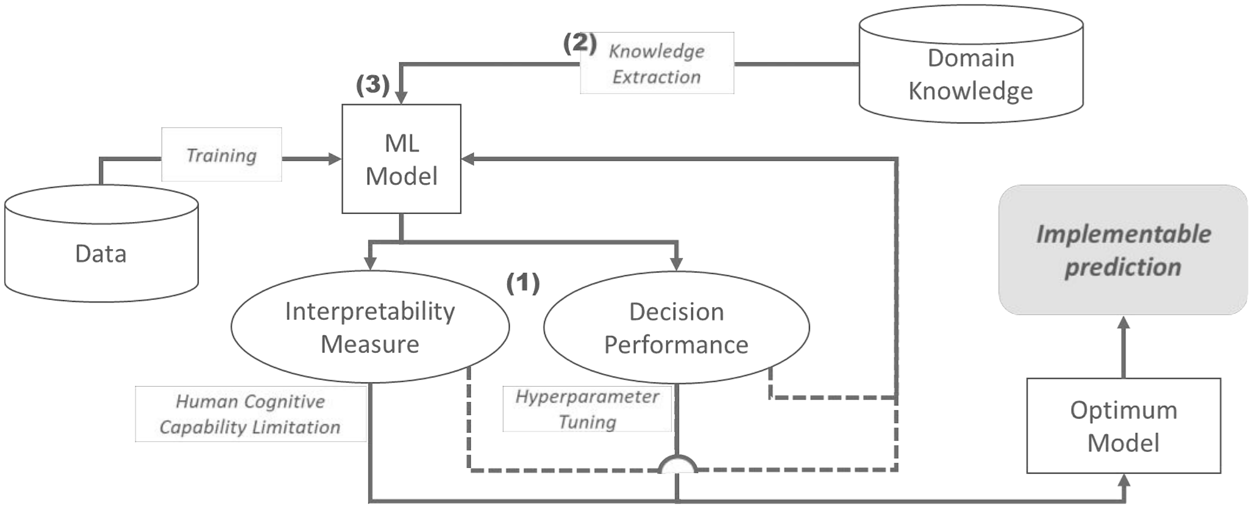


Figure 1. Novel elaborative machine learning framework

Elaborative learning is a cognitive learning approach that utilizes the process of elaboration to improve memory retention and understanding (Willingham, 2013). This method encourages learners to link new data-driven information with prior knowledge. Learners must develop a connection by integrating new information with past knowledge and, by the end, construct the summarization of the comprehensive knowledge. In our framework, the ML model, as the learners, needs to be exposed to previous knowledge, which is the medical domain knowledge in this case. The process of linking this domain knowledge is split into two main steps ([1] and [2] in Figure 1): extracting the knowledge and embedding the knowledge into the ML model. The next step is to verify if the model can provide a good comprehension of learning from the data and domain knowledge by providing an interpretable decision to the human as the evaluator. Thus, we proposed to consider the human cognitive limitation ([3] in Figure 1) in making the summarization. Therefore, the user, as a human collaborator/evaluator, could be sure of the model learning comprehension and the ability to generate robust predictions. In the end, it will build the trust that is necessary for the health-related decision-making process.

3.3. Domain Knowledge Extraction (TF-RCR).

We will incorporate medical domain knowledge in feature selection to instill confidence in the model's prediction powers. Variables mentioned frequently in medical journals will be given high priority in the ML model. The hypothesis is that physicians are familiar with the variables whose associations with the disease have been extensively examined in medical journals. Thus, they will have more confidence about the predictions if the model is built on those variables.

We conducted a structured literature search on PubMed. Articles were retrieved using the keywords “diabetic retinopathy,” “predict,” “factor,” and “determine,” with filters applied to include only clinical studies published between 1990 and 2023. This search yielded a corpus of 13,002 articles, which were subsequently used in the calculation of the TF-RCR (Term Frequency – Relative Citation Ratio) metric. This approach was designed to prioritize features that are both frequently studied and highly cited, ensuring relevance and influence within the medical literature.

The first component is Term Frequency (TF). It comes from the TF/IDF (Term Frequency-Inverse Document Frequency) metric. TF/IDF is a foundational and extensively used methodology in text mining for measuring the word's relevance within a corpus. This strategy is based on the idea that a term's relevance to a document is determined by both its frequency inside that document (Term Frequency) and its uniqueness throughout the entire corpus (Inverse Document Frequency). TF itself is the count of times a word appears in a document. It measures the importance of terms based on their contextual relevance, as demonstrated by Salton and Buckley [12], making it an indispensable tool in a variety of text-mining applications such as information retrieval, document classification, and sentiment analysis.

Research Citation Ratio (RCR) is a bibliometric statistic to assess the relative influence of a scientific work. It is calculated by dividing a paper's citation number by the predicted number of citations based on the journal's citation pattern. It takes into account aspects such as the field's citation practices and the paper's publication age to offer a more thorough assessment of a paper's impact (Hutchins et al., 2016). RCR has gained popularity in academia because of its capacity to account for various citation standards across fields and historical periods, making it a powerful tool for measuring research impact more holistically.

$$\text{TF-RCR} = TF \times RCR \tag{1}$$

In Equation (1), we design the TF-RCR metric by multiplying these two factors to measure and rank the importance of the variables in the domain. We adopted a multiplicative combination of term frequency (TF) and relative citation ratio (RCR) to form the TF-RCR metric, as it provides a simple yet effective way to jointly capture the relevance and impact of medical terms in the literature. TF reflects how frequently a given variable is discussed in the diabetic retinopathy context, while RCR indicates the scientific influence of the publications mentioning that variable. By multiplying TF and RCR, the metric naturally emphasizes features that are both widely discussed and found in high-impact literature, ensuring alignment with both volume and quality of clinical discourse. In contrast, a weighted sum could diminish the relative importance of either dimension and may not sufficiently penalize features that are frequent in low-impact studies or rare in highly cited work. The multiplicative form also maintains scale sensitivity, making it easier to differentiate top-ranked variables, which is critical for feature selection under interpretability constraints.

Algorithm 1. Z-Score Data Normalization

```
Input:
  - data: an array or dataset
Output:
  - normalized_data: an array with CDF values for each data point
Procedure:
  for each data point in data:
      Z-Score = (data point - Mean(data)) / StandardDeviation(data)
      CDF_value = Calculate_Standard_Normal_CDF(Z-Score)
      Add CDF_value to normalized_data
  return normalized_data
```

We normalized the metrics using the commonly used Z-score normalization technique. The transformation helps to normalize the deviation of the metric as the Term frequency becomes extremely big due to the variable counts in the corpus.

3.4. Human Cognitive Capability Limitation.

Miller's theory on human cognitive capacity limitations states that most individuals can efficiently comprehend a maximum of seven variables at a time, with a variance plus or minus two (Miller, 1956).

$$HCCL(x) = \begin{cases} 1 & x \leq 7 \\ \frac{7}{x} & x > 7 \end{cases} \quad where\ x\ is\ \#SelectedFeatures \tag{2}$$

Let x be the number of selected features; we calculate the interpretability score with equation (2) to account for this limitation. The maximum score is obtained when the number of features is fewer than or equal to seven. As the number of features increases, the score gradually decreases, approaching 0.

3.5. Modified Feature Selection (Adaptive RFE).

Recursive Feature Elimination (RFE) is a feature selection approach frequently used in ML to reduce dataset complexity. RFE works by repeatedly deleting the least essential features from the dataset, limiting the collection of predictors to those most influential in explaining the target variable. This procedure is repeated iteratively, with each iteration training the model on a smaller collection of features and judging their relevance using a different estimator. The least significant features are trimmed until the required features are reached or a preset stopping condition is reached (Guyon et al., 2002).

We designed a φ metric to replace accuracy as the determinant metric in choosing the number of features to be adopted into the model. The φ is constructed by multiplying accuracy and the HCCL metric, as both metrics range between 0 and 1. In adaptive RFE, $\Delta\varphi$ is different between the current φ and the previous φ, where the number of selected features is one more compared to the current step. The stopping rule for the RFE loop is updated, and the termination is done if the $\Delta\varphi$ is a threshold that is set at the beginning of the experiment. This threshold is adjusted based on user preferences and how many additional features are allowed. This condition makes our framework customizable to human preferences instead of forcing the user to accept whatever the ML model considers optimal.

Algorithm 2. Adaptive Recursive Feature Elimination

```
Input:
- X: The input matrix (n_samples, n_features).
- y: The target variable.
- estimator: The machine learning model to be used.
- normalized_TF-RCR: The domain knowledge feature importance score
- $\varphi$_treshold: The desired level of interpretability
Output:
- selected_features: The indices of The selected features.
Procedure:
  if n_features_to_select == 1:
    best_feature = select_best_feature(X, y, estimator)
    return [best_feature]
  else:
    current_best_feature = None
    previous_$\varphi$ = -inf
    for feature in range(n_features):
      if feature is not in selected_features:
        new_X = remove_feature(X, feature)
        remaining_features = RFE(new_X, y, estimator, normalized_TF-RCR, n_features_to_select - 1)
        estimator.fit(new_X[:, remaining_features], y)
        HCCL(n_features_to_select-1) = $\begin{cases} 1 & \text{n\_features\_to\_select} - 1 \leq 7 \\ \frac{7}{\text{n\_features\_to\_select}-1} & \text{n\_features\_to\_select} - 1 > 7 \end{cases}$
        $\varphi$ = evaluate(estimator, new_X[:, remaining_features], y)*HCCL
        $\Delta\varphi$ = $\varphi$ - previous_$\varphi$
        if $\Delta\varphi$ > $\varphi$_treshold :
          previous_$\varphi$ = $\varphi$
          current_best_feature = feature
    selected_features.append(current_best_feature)
    return selected_features
```

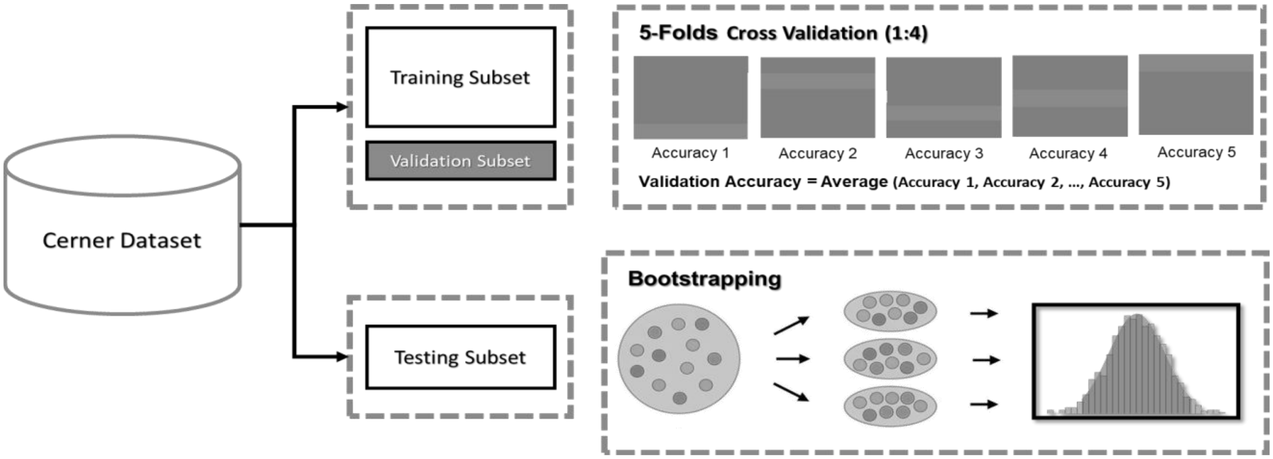


Figure 2. The experiment set-up. (in this setup, we adopted the bootstrapping technique in the testing set)

3.6. Experimental Design.

The experimental setup carefully divides the dataset to facilitate strong model development and unbiased evaluation. Initially, 80% of the dataset is allocated for training and validation, while the remaining 20% is reserved for final testing and remains unchanged throughout the model-building process. This separation method guarantees that the test data is entirely distinct from the training and validation sets, enabling a meaningful assessment of the model's ability to generalize to unknown data.

To achieve greater granularity in the training and validation processes, the original dataset is divided into two subsets: 64% for training and 16% for validation. This internal validation set is essential for fine-tuning model hyperparameters and monitoring training progress. Within the training subset, we implement a 5-fold cross-validation procedure to minimize any biases in model assessment. This involves splitting the training data into five equally sized folds, with each fold serving as the validation set in turn, while the remaining folds are used for training. This iterative procedure is repeated five times, with each fold acting as the validation set once. As a result, this data partitioning approach rigorously evaluates the model's performance, considering both training and validation dynamics. Additionally, it preserves an independent and untouched test set for the final evaluation, in line with best practices in machine learning experimentation.

To test the robustness of the model, we employed a bootstrapping mechanism. Bootstrapping is a resampling approach commonly used in statistical analysis and machine learning to evaluate the robustness and variability of estimators or models. The foundational bootstrap techniques by Efron and Tibshirani gives a complete presentation of the methodology, explaining its foundations and demonstrating its effectiveness across diverse domains (Tibshirani, 1994). The method works by continually pulling random samples from the original dataset with replacement, essentially producing many simulated datasets of the same size as the original. Over several cycles, these resampled datasets are utilized to estimate the means and confidence interval of prediction performance measures. By capturing the variability in the data, bootstrapping provides valuable insights into the stability of the machine learning model's distribution.

3.7. Model Performance Metrics.

In the medical domain, predictive model evaluation frequently relies on a set of performance criteria that are particularly customized to suit the unique difficulties and goals of healthcare applications. Accuracy, sensitivity, specificity, true positive rate (TPR), negative predictive value (NPV), and F1-score are some of the most often used measures. Accuracy, which represents the proportion of properly categorized cases among all occurrences, is a key indicator of a model's overall accuracy. Sensitivity, also known as the true positive rate, assesses a model's ability to properly detect positive cases, which is especially important in situations where false negatives might have profound implications. Specificity, often known as the true negative rate, measures a model's ability to correctly identify negative situations. These measures are critical for thoroughly evaluating medical model diagnostic performance because they provide insight into both the ability to discover actual positive cases and the ability to avoid false alarms. The F1-score, which takes into account both accuracy and recall, gives a balanced assessment of a model's performance, which is especially useful when class distribution is skewed. As described in the literature, particularly the work of (Willingham, 2013), these metrics together serve as key tools in measuring the usefulness and dependability of machine learning models in medical decision-making.

4. Results and Discussion

4.1. Domain Knowledge Extraction.

To extract the existing domain knowledge, we employed a text-mining approach. We measure the importance of each variable based on its term frequency (TF) and the Relative Citation Rate (RCR). Figure 3 displays the initial step of extracting domain knowledge and converting it into importance scores for the variables. We've presented two ranking systems. The first, on the left, only considers the importance of the variable without factoring in the publication's importance to the overall domain knowledge. On the other hand, the one on the right implements our custom metric (TF-RCR) to level the importance of each publication and its contribution to the domain knowledge. The benefit of using TF-RCR instead of TF only is the ability to incorporate the quality of the published article into the variable ranking. The results show no significant differences, indicating that the importance of the variable reaches a consensus.

4.2. Machine Learning Models.

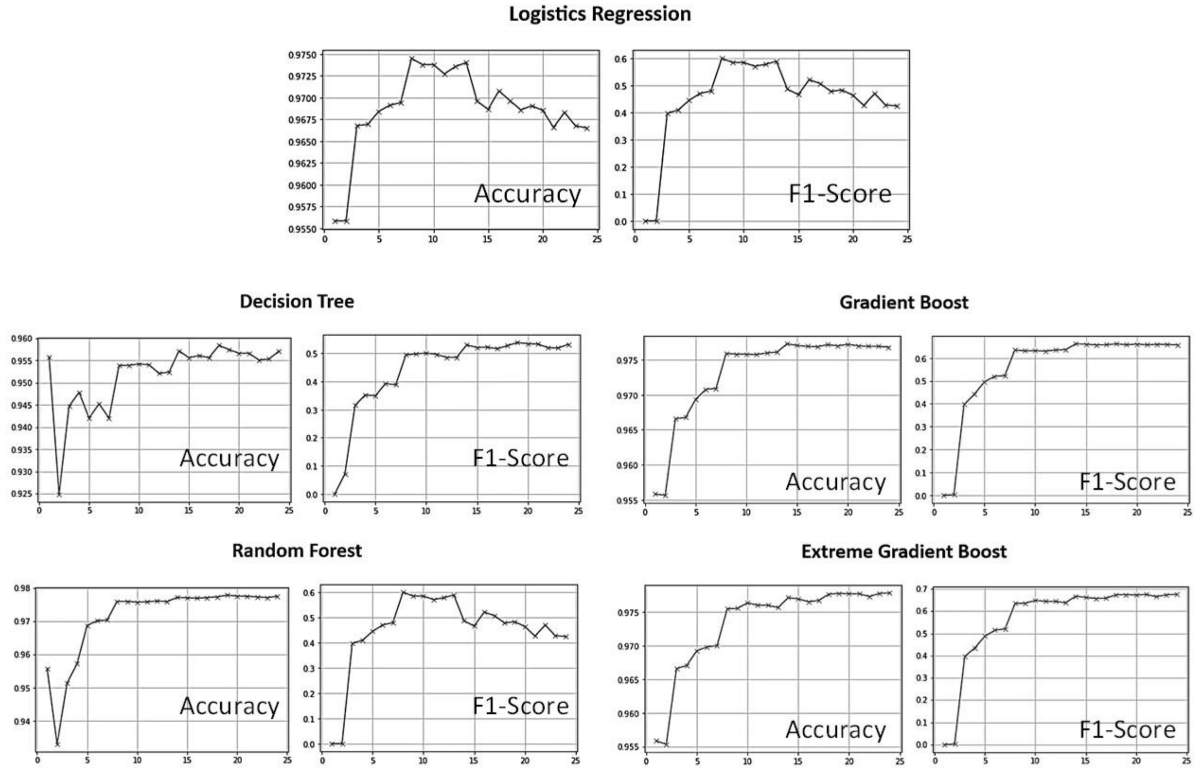


Figure 3. Model ablation result.

Our experimental study focused on five distinct regression or tree-based models: Logistic Regression, Decision Trees, Random Forests, Gradient-Boosting Trees, and Extreme Gradient-Boosting Trees (XGBoost). Those five models are selected because they have higher interpretability than neural-network-based deep learning models. Model performances are evaluated based on their performance metrics, specifically accuracy and F1-score. As delineated in Figure 3, the results indicate an outperformance by the XGBoost model in both accuracy and F1-score compared to its counterparts. The figure also shows a significant accuracy and F1-score drop when we used less than 8 variables for all of the tested models. And vice versa, the metrics do not increase significantly as the number of variables increases beyond 8. Notably, XGBoost not only excelled in these metrics but also demonstrated stability in its performance across various variable sets. This stability is a critical factor in machine learning applications, as it suggests robustness of the variable set and a reliable prediction. Thus, we utilized XGBoost to develop our predictive model based on the elaborative learning framework.

4.3. The DR Prediction Model.

Our proposed framework marks a stride towards addressing the challenge of integrating domain knowledge with a parsimonious yet highly effective model. Our approach culminated in the identification and utilization of a minimal set of eight predictors, which were carefully chosen to align with existing domain knowledge while also ensuring high predictive performance. This number, eight, is particularly noteworthy as it represents an optimal balance between complexity and interpretability. Eight is the number within the interval seven plus minus two as the approved number for a comprehensible number of variables for humans, thereby facilitating easier understanding and application in practical scenarios.

The final predictive model was developed using eight features identified through the TF-RCR metric and clinical relevance criteria: age, glucose, nephropathy (neph), neuropathy (neu), HbA1c, hemoglobin, albumin, and creatinine. These variables consistently appear across patient records and are commonly used in routine diabetic monitoring. Their inclusion ensured not only clinical interpretability but also alignment with practical constraints in real-world deployment. Table 1 presents these features along with their clinical definitions and typical acquisition frequencies. The model achieved strong performance using only these eight inputs, reinforcing the potential of a minimal yet informative feature set in supporting early diabetic retinopathy screening.

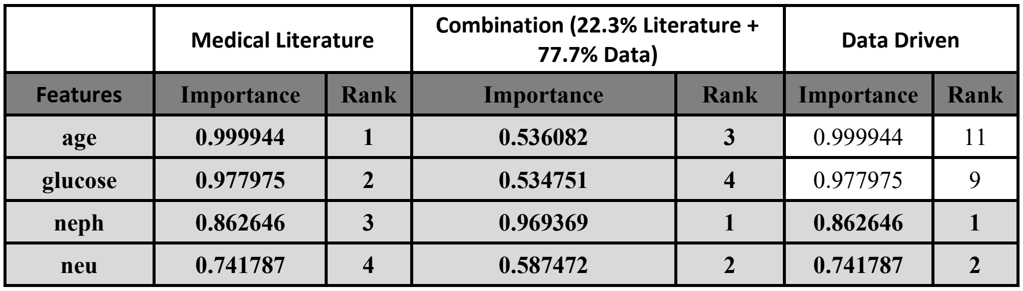
Table 1. The eight DR predictors

Variable	**Clinical Definition**	**Typical Acquisition Frequency**	**Normal Range**
Age	The patient's age is used as a non-modifiable risk factor.	Recorded once; updated only as the patient ages.	n.a. (in years)
Glucose	Blood glucose level (mg/dL), typically measured fasting or randomly to monitor glycemic control.	At every visit, or at least every 3 months.	70–130 [mg/dL]
Nephropathy (Neph)	Diagnosis or clinical indicator of diabetic nephropathy, often inferred from abnormal albuminuria or eGFR.	Evaluated annually albumin-to-creatinine ratio (ACR) and serum creatinine.	0 (1 = positive, 0 = negative)

Variable	**Clinical Definition**	**Typical Acquisition Frequency**	**Normal Range**
Neuropathy (Neu)	Diagnosis of diabetic neuropathy based on clinical symptoms or tests (e.g., monofilament).	Assessed annually through foot exams or symptom screening.	0 (1 = positive, 0 = negative)
HbA1c	Glycated hemoglobin indicates the average blood glucose levels over the past 2 to 3 months, expressed as a percentage.	Every 3–6 months, depending on glycemic control.	< 6.5%
Hemoglobin	The concentration of hemoglobin in blood (g/dL) is a key indicator of anemia and the oxygen-carrying capacity of the blood.	At least annually in diabetic patients, more often in those with renal complications.	Male: 14–18 g/dL Female: 12–16 g/dL
Albumin	Serum albumin level (g/dL) reflects nutritional status and liver/kidney function.	Typically, every 6–12 months or with routine metabolic panels.	3.4–5.4 [g/dL]
Creatinine	Serum creatinine (mg/dL), used to estimate kidney function (eGFR).	At least annually in all diabetic patients.	0.74–1.35 [mg/dL]

Our sensitivity analysis shows the robustness and relevance of these eight variables. By contrasting the model's performance when informed by a comprehensive set of 100% variable rankings derived from the literature against a configuration where the influence is predominantly data-driven (22.3% literature-informed and 77.7% data-derived), as shown in table 2, we observe a consistency in the selection of these eight variables. This consistency is not trivial; it indicates that a substantial proportion (77.7%) of the variation in our dataset, which is the source of our model development, is in concordance with established domain knowledge. Such alignment provides a substantial boost in confidence regarding the model's applicability and validity, as it implies that the model is not only data-driven but also grounded in and corroborated by domain-specific expertise.

Table 2. Medical-Literature-Based vs Data-Based feature selection



	Medical Literature		**Combination (22.3% Literature + 77.7% Data)**		**Data Driven**	
Features	**Importance**	**Rank**	**Importance**	**Rank**	**Importance**	**Rank**
age	**0.999944**	**1**	**0.536082**	**3**	0.999944	11
glucose	**0.977975**	**2**	**0.534751**	**4**	0.977975	9
neph	**0.862646**	**3**	**0.969369**	**1**	**0.862646**	**1**
neu	**0.741787**	**4**	**0.587472**	**2**	**0.741787**	**2**

	Medical Literature		Combination (22.3% Literature + 77.7% Data)		Data Driven	
Features	**Importance**	**Rank**	**Importance**	**Rank**	**Importance**	**Rank**
hba1c	**0.502284**	**5**	**0.48835**	**5**	**0.502284**	**3**
hemoglobin	**0.497377**	**6**	**0.44612**	**6**	**0.497377**	**6**
albumin	**0.481435**	**7**	**0.418853**	**8**	0.481435	14
creatinine	**0.377927**	**8**	**0.443007**	**7**	**0.377927**	**4**
sodium	0.370736	9	0.393718	12	0.370736	15
calcium	0.338984	10	0.385894	15	0.338984	17
triglyceride	0.330657	11	0.38619	14	0.330657	13
bilirubin	0.326602	12	0.38309	17	0.326602	18
rbc	0.323919	13	0.38306	18	0.323919	16
wbc	0.322088	14	0.400295	10	**0.322088**	**7**
ast	0.321929	15	0.379821	21	0.321929	21
chloride	0.321089	16	0.38174	19	0.321089	19
potassium	0.320928	17	0.380954	20	0.320928	20
hematocrit	0.319857	18	0.418822	9	**0.319857**	**5**
bun	0.318996	19	0.396505	11	**0.318996**	**8**
alt	0.31792	20	0.383887	16	0.31792	12
anion_gap	0.317293	21	0.387329	13	0.317293	10
mch	0.317009	22	0.378484	22	0.317009	22
mchc	0.317009	22	0.378077	23	0.317009	23
mcv	0.317009	22	0.377621	24	0.317009	24

Our model's resilience and performance have been highlighted by its training process, which adopts the 5-fold cross-validation procedure. Throughout the validation process, we have seen that all performance indicators are constantly steady at 95%, as shown in Table 3. This degree of consistency attests to the model's ability to uncover important patterns and generalize from the input data.

Table 3. Model optimization result.

Evaluation Metric	**Fold #1**	**Fold #2**	**Fold #3**	**Fold #4**	**Fold #5**	**Mean**	**Confidence Interval**	
Accuracy	0.967	0.963	0.964	0.966	0.964	0.965	0.963	0.967
Specificity	0.983	0.980	0.978	0.980	0.983	0.981	0.978	0.983
Sensitivity	0.951	0.947	0.951	0.952	0.945	0.949	0.945	0.953
AUROC	0.967	0.963	0.964	0.966	0.964	0.965	0.963	0.967
PPV	0.982	0.979	0.977	0.980	0.982	0.980	0.978	0.983
NPV	0.953	0.948	0.952	0.953	0.946	0.950	0.947	0.954
F1	0.966	0.963	0.964	0.966	0.963	0.964	0.962	0.966

4.4. Model Robustness.

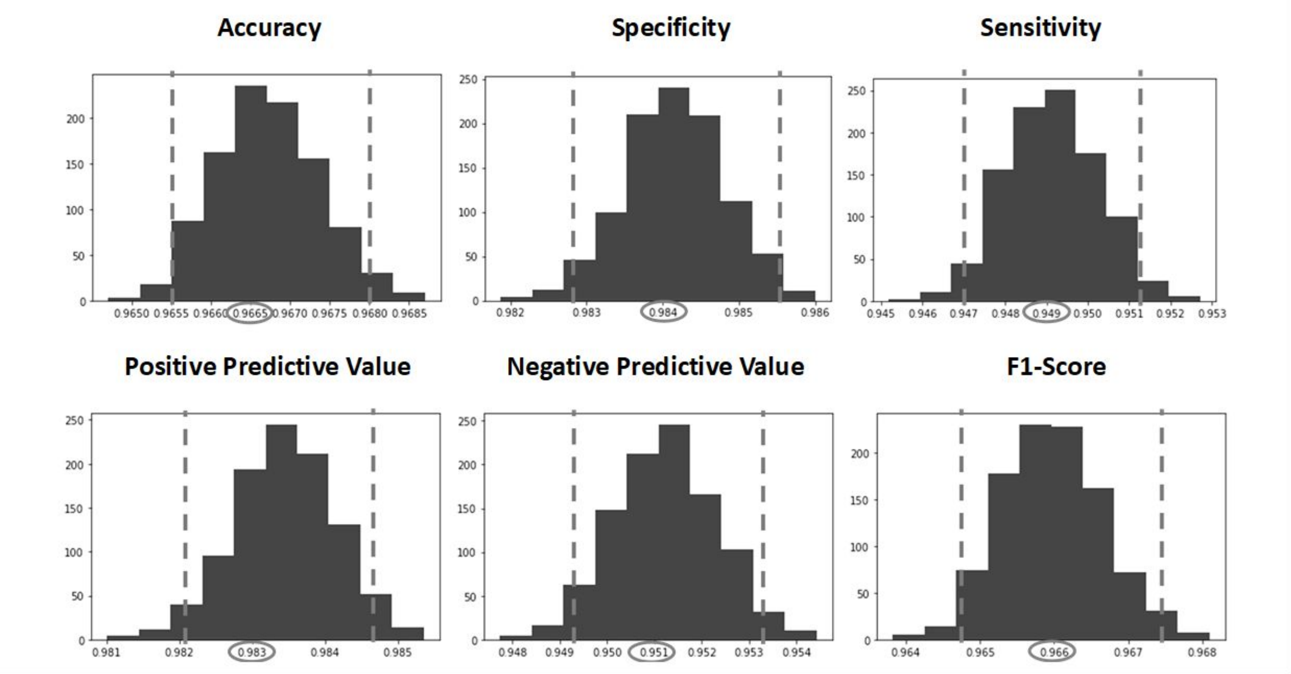


Figure 4. Optimum model performance over simulated new instances (using bootstrapped testing data)

All the previous results provide evidence that our optimum model is well-suited to the past data and is supported by guidance from domain knowledge. However, it will be useless if we cannot support our optimum model to perform well in predicting future instances. To confirm this notion, the final step of our experiment simulates the new dataset, and as shown in Figure 4, all performance metrics are above 95% on average. Over a thousand new instances, the lowest predicting power is above 94%. This implies that our optimum model is robust for predicting future instances.

Furthermore, Table 4 summarizes the performance of both image-based and non-image-based models for diabetic retinopathy (DR) prediction. The upper section reports previously published results from retinal image-based classifiers (Kumar & Madheswaran, 2012), including Support Vector Machine (SVM), Backpropagation Neural Network (BPN), Adaptive Neuro-Fuzzy Inference System (ANFIS), K-Nearest Neighbors (KNN), and Learning Vector Quantization (LVQ). These models generally exhibit high sensitivity (ranging from 0.964 to 0.978) but variable specificity, with values as low as 0.532. While image-based models are well-established in DR screening and constitute the majority of FDA-approved tools, they depend on specialized imaging infrastructure and clinical workflows that may not be readily accessible in resource-limited or primary care settings.

The lower section of Table 4 presents results from models trained on structured, laboratory-based tabular data. These include traditional machine learning algorithms (logistic regression, random forest, XGBoost) and deep learning architectures (feedforward neural network, Temporal Convolutional Network (TCN)) (Wang et al., 2024), and a recent self-attentive temporal model (MB-TCN-TC-10). Among these, MB-TCN-TC-10 achieved the highest accuracy (0.983) and specificity (0.991), although its sensitivity (0.734) was comparatively lower, which may increase the risk of missed DR cases in clinical screening.

Table 4. Model comparison.

Model Name	Accuracy	Specificity	Sensitivity	PPV	NPV	F1	AUROC
Image-Based Model (Kumar & Madheswaran, 2012)							
SVM	0.975	0.892	0.978	n.a.	n.a.	n.a.	n.a.
BPN	0.948	0.674	0.968	n.a.	n.a.	n.a.	n.a.
ANFIS	0.939	0.532	0.964	n.a.	n.a.	n.a.	n.a.
KNN	0.948	0.683	0.969	n.a.	n.a.	n.a.	n.a.
LVQ	0.964	0.807	0.973	n.a.	n.a.	n.a.	n.a.
Non-Image-Based Model (Wang et al., 2024)							
MB-TCN-TC-10	0.983	0.991	0.734	0.723	0.992	0.728	0.949
Original TCN	0.972	0.981	0.703	0.539	0.99	0.61	0.893
Neural network	0.934	0.941	0.719	0.28	0.991	0.403	0.9
Random forest	0.962	0.97	0.705	0.431	0.99	0.535	0.907
XGBoost	0.963	0.971	0.708	0.436	0.991	0.54	0.915
Logistic regression	0.958	0.967	0.692	0.399	0.99	0.506	0.882
Our Proposed Method							
Elaborative-XGBoost	**0.965**	**0.981**	**0.949**	**0.98**	**0.95**	**0.964**	**0.965**

The proposed Elaborative-XGBoost model achieved a balanced and clinically favorable performance profile, with an accuracy of 0.965, specificity of 0.981, and sensitivity of 0.949. It also demonstrated strong positive predictive value (PPV = 0.980), negative predictive value (NPV = 0.950), F1-score (0.964), and AUROC (0.965). Notably, the model retains a transparent and inherently interpretable structure aligned with clinician reasoning, facilitating validation and adoption in routine care. While recent advances have improved post-hoc interpretability in deep models, the capacity of inherently interpretable algorithms to directly support clinician trust and real-time decision-making remains essential, particularly in high-stakes and resource-constrained environments.

4.5. Model Interpretability.

To illustrate how our model produces interpretable outputs, we provide a representative patient-level explanation using SHAP (SHapley Additive exPlanations) values. Figure 5a presents a summary plot of SHAP values for the 8-feature model. Each dot represents a SHAP value for a feature and an instance, colored by feature value (red = high, blue = low). Features like creatinine, hba1c, age, and hemoglobin exhibit the strongest contributions, with clear patterns indicating that high creatinine and hba1c increase the predicted risk, while low hemoglobin and higher age tend to reduce it. This visualization helps clinicians understand which features drive the model's overall decision-making across the population.

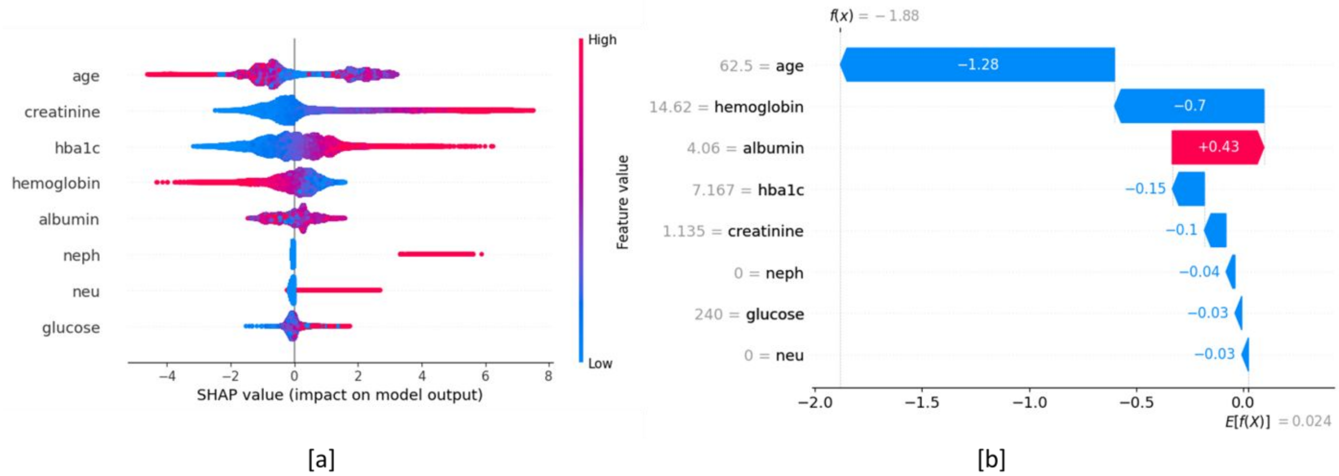


Figure 5. SHAP explanation for 8 selected features.

In Figure 5b, we provide an individual force plot-style explanation (SHAP waterfall chart) for a single prediction. This shows how the model's output is constructed by aggregating the contribution of each feature from the base value (average model output). For the selected patient, the prediction is decreased primarily due to older age and low hemoglobin, while moderately elevated albumin contributes a slight increase in risk. This decomposition offers clinicians a transparent view into why a patient was classified at a particular risk level, without requiring auxiliary explanation tools.

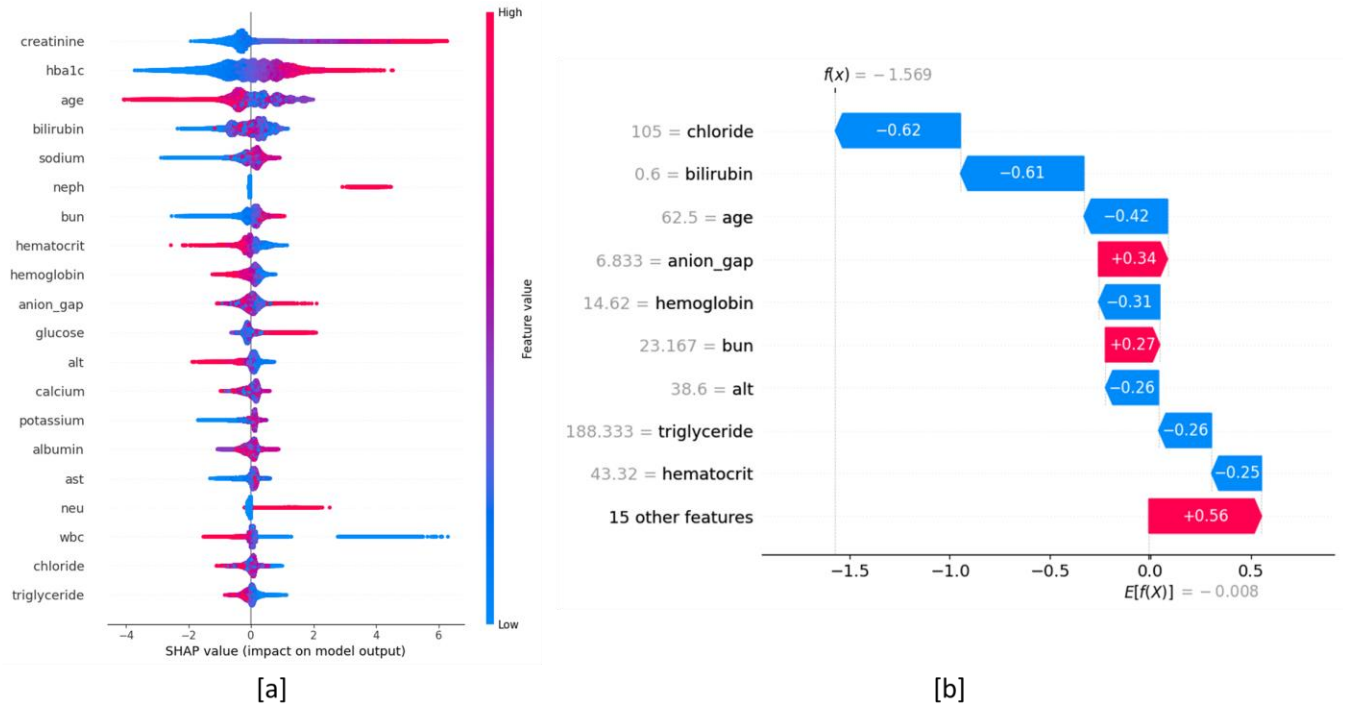


Figure 6. SHAP explanation for 24 selected features.

To compare interpretability under cognitive constraints, Figures 6a and 6b replicate the SHAP summary and individual explanation plots for the full 24-feature model. While performance increased marginally, by less than 1% in AUC, this came at the cost of reduced clarity. As shown in Figure 6a, the expanded feature space introduces dense overlapping patterns with many low-impact variables, such as WBC, triglyceride, and chloride, making it difficult to discern the primary drivers of prediction. Figure 6b illustrates a corresponding individual prediction explanation. Although similar risk drivers, bilirubin and anion gap, still play a role, the influence is now fragmented across a broader range of features, hindering interpretability.

Overall, the 8-feature model preserves nearly the same predictive performance while offering more concise and cognitively accessible explanations. This aligns with the goals of our elaborative learning framework, where interpretability and model transparency are prioritized alongside predictive accuracy. These findings support the use of constrained, domain-informed feature sets to produce trustable decision support tools in clinical settings.

5. Conclusion

In conclusion, this study has effectively demonstrated the implementation of a novel comprehensive learning approach tailored to harness medical domain knowledge for enhancing the explainability of AI in detecting diabetic retinopathy. Achieving an impressive 96.65% accuracy and a 96.60% F1-Score with only eight features, the model not only adheres to the constraints of human cognitive processing capacities but also seamlessly aligns with medical expert understanding. This dual achievement underscores the model's potential to serve as an effective screening tool in primary care settings, where early detection of sight-threatening conditions is crucial. Furthermore, the use of a limited number of features, each deeply rooted in established medical knowledge, ensures that the model remains both practical and relevant to everyday clinical practices, bridging the gap between advanced AI technologies and their real-world clinical applications.

Moreover, the integration of medical domain expertise in the AI model's learning process represents a significant stride towards overcoming the longstanding barriers of machine learning explainability in clinical settings. The model's robust performance, coupled with its accountability and transparency, positions it as a reliable and valuable asset in the medical community. As AI continues to evolve, the methodology adopted in this study offers a scalable template for developing future disease prediction models that are not only accurate but also interpretable and user-friendly for medical professionals. This approach paves the way for broader acceptance and integration of AI technologies in healthcare, ultimately enhancing patient outcomes through more informed and timely medical decision-making.

Although the dataset used in this study is derived from the CERNER EHR system, which encompasses a wide network of hospitals and healthcare providers across the United States, we acknowledge that external validation remains essential to further establish the model’s generalizability. While the current data offers substantial coverage of diverse patient populations and clinical practices, validating the model on an independent dataset would allow us to assess its robustness across different institutional settings, data collection protocols, and patient demographics. We consider this an important direction for future research to ensure broader applicability and clinical reliability of the proposed framework. Currently, we are in the process to obtain the access to another reliable large dataset for external validation.

The current study is based on static, cross-sectional laboratory measurements, which may not fully capture the progression dynamics of diabetic retinopathy (DR). While methods for temporal modeling, like recurrent neural networks (RNNs) and temporal convolutional networks (TCNs), have proven effective in capturing long-term trends in biomarkers such as HbA1c, this paper aims to develop models that are straightforward, easy to understand, and in line with clinical reasoning. Prior work by other members of our research team has explored such temporal models in greater depth, including a multi-branching TCN with tensor data completion for DR prediction (Wang et al., 2024) and deep learning on longitudinal EHRs (Chen et al., 2022). While these models demonstrate strong performance, our current framework achieves comparable predictive accuracy while offering greater interpretability, which is crucial for clinical adoption. Future extensions of this work may integrate temporal modeling to enhance personalization and time-sensitive risk assessment, while maintaining the model’s transparency.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Institutes of Health SCH: Harnessing Tensor Information to Improve EHR Data Quality for Accurate Data-driven Screening of Diabetic Retinopathy with Routine Lab Results, Grant Number: 1R01EY033861-01.

References

Abedin, B. (2021). Managing the tension between opposing effects of explainability of artificial intelligence: A contingency theory perspective. *Internet Research*, *32*(2), 425–453. https://doi.org/10.1108/INTR-05-2020-0300

Amann, J., Vetter, D., Blomberg, S. N., Christensen, H. C., Coffee, M., Gerke, S., Gilbert, T. K., Hagendorff, T., Holm, S., Livne, M., Spezzatti, A., Strümke, I., Zicari, R. V., Madai, V. I., & Initiative, on behalf of the Z.-I. (2022). To explain or not to explain?—Artificial intelligence explainability in clinical decision support systems. *PLOS Digital Health*, *1*(2), e0000016. https://doi.org/10.1371/journal.pdig.0000016

Arbelaez Ossa, L., Starke, G., Lorenzini, G., Vogt, J. E., Shaw, D. M., & Elger, B. S. (2022). Re-focusing explainability in medicine. *DIGITAL HEALTH*, *8*, 20552076221074488. https://doi.org/10.1177/20552076221074488

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1721–1730. https://doi.org/10.1145/2783258.2788613

Chen, S., Wang, Z., Yao, B., & Liu, T. (2022). Prediction of Diabetic Retinopathy Using Longitudinal Electronic Health Records. *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 949–954. https://doi.org/10.1109/CASE49997.2022.9926605

Diabetic Retinopathy Tables | National Eye Institute. (n.d.). Retrieved October 27, 2023, from https://www.nei.nih.gov/learn-about-eye-health/eye-health-data-and-statistics/diabetic-retinopathy-data-and-statistics/diabetic-retinopathy-tables

Fisher, M. D., Rajput, Y., Gu, T., Singer, J. R., Marshall, A. R., Ryu, S., Barron, J., & MacLean, C. (2016). Evaluating Adherence to Dilated Eye Examination Recommendations Among Patients with Diabetes, Combined with Patient and Provider Perspectives. *American Health & Drug Benefits*, *9*(7), 385.

Flaxel, C. J., Adelman, R. A., Bailey, S. T., Fawzi, A., Lim, J. I., Vemulakonda, G. A., & Ying, G. (2020). Diabetic Retinopathy Preferred Practice Pattern®. *Ophthalmology*, *127*(1), P66–P145. https://doi.org/10.1016/j.ophtha.2019.09.025

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, *46*(1), 389–422. https://doi.org/10.1023/A:1012487302797

Holzinger, A., Langs, G., Denk, H., Zatloukal, K., & Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. *WIREs Data Mining and Knowledge Discovery*, *9*(4), e1312. https://doi.org/10.1002/widm.1312

Hutchins, B. I., Yuan, X., Anderson, J. M., & Santangelo, G. M. (2016). Relative Citation Ratio (RCR): A New Metric That Uses Citation Rates to Measure Influence at the Article Level. *PLOS Biology*, *14*(9), e1002541. https://doi.org/10.1371/journal.pbio.1002541

Improving Students' Learning With Effective Learning Techniques: Promising Directions From Cognitive and Educational Psychology—John Dunlosky, Katherine A. Rawson, Elizabeth J. Marsh, Mitchell J. Nathan, Daniel T. Willingham, 2013. (n.d.). Retrieved October 4, 2023, from https://journals.sagepub.com/doi/10.1177/1529100612453266

Kovarik, J. J., Eller, A. W., Willard, L. A., Ding, J., Johnston, J. M., & Waxman, E. L. (2016). Prevalence of undiagnosed diabetic retinopathy among inpatients with diabetes: The diabetic retinopathy inpatient study (DRIPS). *BMJ Open Diabetes Research and Care*, *4*(1), e000164. https://doi.org/10.1136/bmjdrc-2015-000164

Kumar, S. J. J., & Madheswaran, M. (2012). An Improved Medical Decision Support System to Identify the Diabetic Retinopathy Using Fundus Images. *Journal of Medical Systems*, *36*(6), 3573–3581. https://doi.org/10.1007/s10916-012-9833-3

Lee, D., & Yoon, S. N. (2021). Application of Artificial Intelligence-Based Technologies in the Healthcare Industry: Opportunities and Challenges. *International Journal of Environmental Research and Public Health*, *18*(1), Article 1. https://doi.org/10.3390/ijerph18010271

Levin, J. R. (1988). Elaboration-based learning strategies: Powerful theory = powerful application. *Contemporary Educational Psychology*, *13*(3), 191–205. https://doi.org/10.1016/0361-476X(88)90020-3

Lundberg, S. M., Nair, B., Vavilala, M. S., Horibe, M., Eisses, M. J., Adams, T., Liston, D. E., Low, D. K.-W., Newman, S.-F., Kim, J., & Lee, S.-I. (2018). Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, *2*(10), Article 10. https://doi.org/10.1038/s41551-018-0304-0

Markus, A. F., Kors, J. A., & Rijnbeek, P. R. (2021). The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, *113*, 103655. https://doi.org/10.1016/j.jbi.2020.103655

Mersha, G. A., Alimaw, Y. A., & Woredekal, A. T. (2022). Prevalence of diabetic retinopathy among diabetic patients in Northwest Ethiopia—A cross sectional hospital based study. *PLOS ONE*, *17*(1), e0262664. https://doi.org/10.1371/journal.pone.0262664

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, *63*, 81–97. https://doi.org/10.1037/h0043158

Mrugacz, M., Bryl, A., & Zorena, K. (2021). Retinal Vascular Endothelial Cell Dysfunction and Neuroretinal Degeneration in Diabetic Patients. *Journal of Clinical Medicine*, *10*(3), Article 3. https://doi.org/10.3390/jcm10030458

National Diabetes Statistics Report 2020. Estimates of diabetes and its burden in the United States. (2020).

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., Sun, M., Sundberg, P., Yee, H., Zhang, K., Zhang, Y., Flores, G., Duggan, G. E., Irvine, J., Le, Q., Litsch, K., … Dean, J. (2018). Scalable and accurate deep learning with electronic health records. *Npj Digital Medicine*, *1*(1), Article 1. https://doi.org/10.1038/s41746-018-0029-1

Retinopathy in Diabetes | Diabetes Care | American Diabetes Association. (n.d.). Retrieved October 27, 2023, from https://diabetesjournals.org/care/article/27/suppl_1/s84/24669/Retinopathy-in-Diabetes

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

Shailaja, K., Seetharamulu, B., & Jabbar, M. A. (2018). Machine Learning in Healthcare: A Review. *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 910–914. https://doi.org/10.1109/ICECA.2018.8474918

Tibshirani, B. E., R. J. (1994). *An Introduction to the Bootstrap*. Chapman and Hall/CRC. https://doi.org/10.1201/9780429246593

Ting, D. S. W., Cheung, C. Y.-L., Lim, G., Tan, G. S. W., Quang, N. D., Gan, A., Hamzah, H., Garcia-Franco, R., San Yeo, I. Y., Lee, S. Y., Wong, E. Y. M., Sabanayagam, C., Baskaran, M., Ibrahim, F., Tan, N. C., Finkelstein, E. A., Lamoureux, E. L., Wong, I. Y., Bressler, N. M., … Wong, T. Y. (2017). Development and Validation of a Deep Learning System for Diabetic Retinopathy and Related Eye Diseases Using Retinal Images From Multiethnic Populations With Diabetes. *JAMA*, *318*(22), 2211–2223. https://doi.org/10.1001/jama.2017.18152

Wang, F., Casalino, L. P., & Khullar, D. (2019). Deep Learning in Medicine—Promise, Progress, and Challenges. *JAMA Internal Medicine*, *179*(3), 293–294. https://doi.org/10.1001/jamainternmed.2018.7117

Wang, X., & Yin, M. (2021). Are Explanations Helpful? A Comparative Study of the Effects of Explanations in AI-Assisted Decision-Making. *26th International Conference on Intelligent User Interfaces*, 318–328. https://doi.org/10.1145/3397481.3450650

Wang, Z., Chen, S., Liu, T., & Yao, B. (2024). Multi-Branching Temporal Convolutional Network With Tensor Data Completion for Diabetic Retinopathy Prediction. *IEEE Journal of Biomedical and Health Informatics*, *28*(3), 1704–1715. https://doi.org/10.1109/JBHI.2024.3351949

Westerlund, A. M., Hawe, J. S., Heinig, M., & Schunkert, H. (2021). Risk Prediction of Cardiovascular Events by Exploration of Molecular Data with Explainable Artificial Intelligence. *International Journal of Molecular Sciences*, *22*(19), Article 19. https://doi.org/10.3390/ijms221910291

Yau, J. W. Y., Rogers, S. L., Kawasaki, R., Lamoureux, E. L., Kowalski, J. W., Bek, T., Chen, S.-J., Dekker, J. M., Fletcher, A., Grauslund, J., Haffner, S., Hamman, R. F., Ikram, M. K., Kayama, T., Klein, B. E. K., Klein, R., Krishnaiah, S., Mayurasakorn, K., O'Hare, J. P., … for the Meta-Analysis for Eye Disease (META-EYE) Study Group. (2012). Global Prevalence and Major Risk Factors of Diabetic Retinopathy. *Diabetes Care*, *35*(3), 556–564. https://doi.org/10.2337/dc11-1909

Sample Paper 2 - Temporal Neural Network Driven Demand Forecasting

Original Paper

Advanced Temporal based Neural Network Model for Blood Components Demand Forecasting

Abstract

Background: Accurate blood demand forecasting is essential for ensuring an efficient and reliable blood supply, as each blood component has a limited shelf life. Inaccurate forecasting can lead to critical shortages during emergencies, putting patients at risk, or excessive wastage due to unused blood stocks. Addressing these challenges requires advanced models capable of capturing complex demand patterns and responding effectively to sudden fluctuations.

Objective: This study aims to enhance blood demand forecasting performance by leveraging two deep learning techniques, specifically Temporal Convolutional Network (TCN) and Long Short-Term Memory (LSTM).

Methods: TCN and LSTM are applied to forecast the demand for three blood components, namely Fresh Frozen Plasma (FFP), Red Blood Cells (RBC), and Thrombocyte Concentrate (TC). Their performance is compared against traditional forecasting methods, including ARIMA, XGBoost, and Neural Networks (NN), to validate their effectiveness.

Results: The results demonstrate that TCN and LSTM consistently outperform traditional models across all blood components. For FFP, TCN demonstrated the best performance with an average MAE of 4.594 and an average RMSE of 7.266. For RBC, LSTM demonstrated the best performance, achieving an average MAE of 12.345 and an average RMSE of 15.221. For TC, LSTM demonstrated the ability to capture general demand patterns with a lower average MAE of 6.863 and average of RMSE of 9.028. This result indicates a better overall performance and its strength in minimizing larger errors and modeling gradual temporal changes. However, both models show limitations in accurately capturing sudden spikes, suggesting that further refinement is needed to improve their responsiveness to sudden spikes without compromising overall forecasting performance.

Conclusions: The findings of this study highlight the potential of deep learning techniques, particularly Temporal Convolutional Networks (TCN) and Long Short-Term Memory (LSTM), to improve the accuracy of blood demand forecasting. Compared to traditional and machine learning methods such as ARIMA, artificial neural networks (ANN), and XGBoost, these models demonstrate superior performance in capturing complex temporal patterns. Their ability to model complex temporal dynamics offers a valuable advantage in managing short-lived medical resources such as blood products. Future research may explore integrate modeling approaches and address the challenge of extreme demand events.

Keywords: blood demand; forecasting; Long Short-Term Memory (LSTM); Temporal Convolutional Network (TCN)

Introduction

As an important component of the human body, blood plays a vital role in the anatomical functions that consist of several components, mainly red blood cells, platelets, and blood plasma.[1–3] Red blood cells can help manage hemorrhage and enhancing oxygen transmission to tissues, fresh frozen plasma can be utilized to counteract the effects of anticoagulants, platelets use to prevent hemorrhage in patients with thrombocytopenia, and cryoprecipitate can be used in hypofibrinogenemia case.[3] Given its essential function and the specific roles of each blood component, efficient blood product management is crucial and needs careful consideration, particularly in ensuring that supply meets demand during surgeries or emergency situations.[4]

However, managing blood supply presents several challenges. First, excessive blood orders can lead to inefficiencies in time, resources, and cost, as highlighted by previous research.[5] Second, blood products have maximum shelf-life criteria, namely Red Blood Cells last around 40 days with platelets having survival up to 7 days after leaving the human body.[6,7] Given its perishable nature, accurate demand forecasting is crucial to minimizing wastage while ensuring a sufficient blood supply. Furthermore, preventing critical shortage is essential to avoid putting patients at risk. To address these challenges, prior research has explored various forecasting methods to manage the uncertainty of blood demand effectively.

Initially, research focused on time series forecasting models. Shih and Rajendran[8] found that traditional time series models, ARIMA, outperformed machine learning methods in practicing blood demand. Around the same time, Fanoodi et al[9] applied ANN and ARIMA models to forecast daily blood requests, emphasizing the importance of effective demand forecasting in preventing shortages.

Subsequent studies introduced machine learning techniques for more accurate forecasting across different blood components. Moslemi and Attari[10] utilized ANN to predict monthly blood demand based on various blood products and groups. Li et al[11] combined statistical time series modelling, machine learning, and operation research techniques to optimize red blood cells demand forecasting. Meanwhile, Sun et al[12] applied the XGBoost model to analyze daily and weekly red blood cell demand, demonstrating improved trend recognition.

Further advancements focused on enhancing accuracy and addressing demand uncertainties. Elmir et al[13] explored machine learning and time series forecasting methods to enhance monthly blood demand forecast, improving supply chain efficiency and reducing waste. More recently, Wang et al[14] utilized SARIMAX and LSTM models to forecast daily blood demand, highlighting the growing importance of deep learning techniques in this field.

Unlike red blood cells, platelets pose an even greater challenge in demand forecasting due to their high cost and extremely short shelf life. Their usage varies significantly, requiring advanced predictive techniques. To tackle this, Motamedi et al[15] developed an efficient platelet demand forecasting model utilizing ARIMA, Prophet, Lasso Regression, Random Forest, and LSTM to predict daily platelet demand. Their study demonstrated that LSTM outperformed other models in predicting daily platelet demand.

Prior studies have explored various methods for blood demand forecasting. However, despite these efforts, the complexity of sequential dependencies and temporal variations in demand continues to pose a significant challenge. Among the most promising deep learning models for time series forecasting is Long-Short Term Memory (LSTM). LSTM is a modified version of Recurrent Neural Networks (RNN) designed to overcome the "vanishing gradient" problem, which frequently occurs in RNNs.[16] It selectively retains or discard information, allowing it to capture long-term dependencies effectively. These characteristics make LSTM particularly advantageous for processing, forecasting, and classifying time series data. Compared to ANN, LSTM has demonstrated superior performance in forecasting tasks.[17,18]

While LSTM is widely used for sequential data modeling, Temporal Convolutional Network (TCN) has emerged as a competitive alternative with superior performance in some forecasting applications. Unlike LSTM, TCN leverages causal and dilated convolutions, enabling parallel sequence processing and more efficient long-range dependency capture. Pei[19] found that TCN results in a better performance than LSTM. Several studies have also demonstrated TCN's effectiveness, including Ghimire et al[20] in modeling electricity demand uncertainty and Bernacki[21] in air pollution concentrations forecasting. These findings indicate the potential of leveraging Temporal Convolutional Networks (TCNs) to enhance performance in time series forecasting, making them a promising approach for blood demand forecasting.

Given the strengths of both LSTM and TCN, this study aims to enhance blood demand forecasting accuracy by leveraging these advanced deep learning techniques. Rather than merely comparing different methods, this research aims to leverage the strengths of both LSTM and TCN models to enhance forecasting performance, focusing on their capabilities to capture temporal dependencies. Benchmarking against previous models will be conducted to validate their effectiveness in blood demand forecasting.

Methods

Material

The data used in this study consist of historical blood demand collected by Indonesia Red Cross Society or Palang Merah Indonesia, DKI Jakarta from January 2020 until March 2023. There are three main components of blood products provided by the data, namely Red Blood Cells (RBC) or red blood cells, Fresh Frozen Plasma (FFP) or Blood Plasma, and Thrombocyte Concentrate (TC) or Platelets. The data structure consists of dates, blood components, blood types, and the number of requests. Forecasting models will be made in daily terms for each component with its blood type (A, B, AB and O) and will be used for comparing each performance.

Data use was permitted under an official request letter, which served as the formal authorization for data access and use. The dataset contained no personal or identifiable information, and is available from the corresponding author upon reasonable request, subject to approval by the Indonesian Red Cross.

Experiment Design

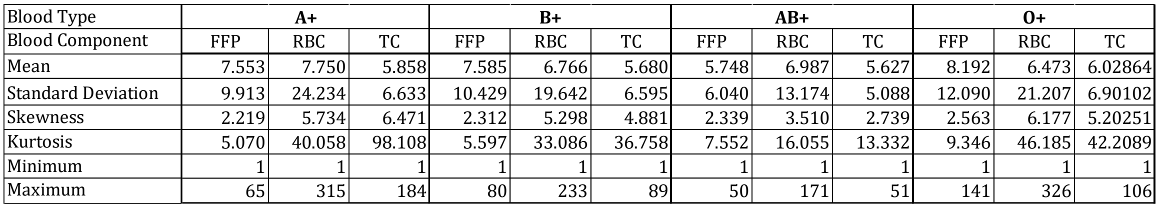
The data used shows that blood groups with negative rhesus for each component have intermittent demand patterns with very high demand interval variations. To avoid high model errors, blood demand data for negative rhesus blood groups is not used in data processing. No normalization or imputation was applied, as the dataset was used in its raw form to preserve the original demand patterns required for the modeling objectives, and the dataset contained no missing value.

The study focuses on forecasting the demand for three primary blood components: RBC, FFP, and TC, to get a robust comparison. These components were selected because they are the most used in transfusions. Furthermore, combining the three blood components with the four major blood types (A+, B+, AB+, and O+) results in 12 demand models, ensuring comprehensive blood demand scenarios. The design enables an in-depth analysis of forecasting performance across varying demand on each component and its type, which can ensure that the results are applicable to diverse operational conditions in blood management.

Descriptive Statistics

Descriptive statistics were calculated to better understand demand characteristics for each blood type and component. The metrics reported include the mean, standard deviation, skewness, kurtosis, minimum, and maximum observed values. These statistics provide an overview of the central tendency, variability, and distribution shape of the dataset prior to any further analysis.

Table 1. Descriptive statistics of blood demand by type and component



Blood Type	A+			B+			AB+			O+		
Blood Component	FFP	RBC	TC	FFP	RBC	TC	FFP	RBC	TC	FFP	RBC	TC
Mean	7.553	7.750	5.858	7.585	6.766	5.680	5.748	6.987	5.627	8.192	6.473	6.02864
Standard Deviation	9.913	24.234	6.633	10.429	19.642	6.595	6.040	13.174	5.088	12.090	21.207	6.90102
Skewness	2.219	5.734	6.471	2.312	5.298	4.881	2.339	3.510	2.739	2.563	6.177	5.20251
Kurtosis	5.070	40.058	98.108	5.597	33.086	36.758	7.552	16.055	13.332	9.346	46.185	42.2089
Minimum	1	1	1	1	1	1	1	1	1	1	1	1
Maximum	65	315	184	80	233	89	50	171	51	141	326	106

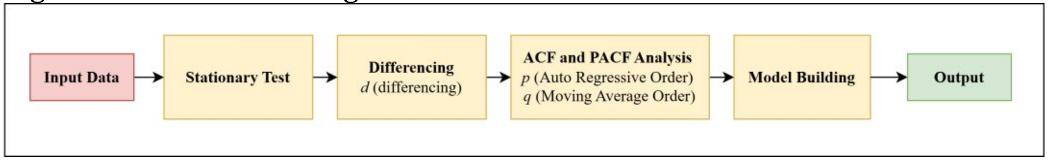
Table 1 shows demand variability across blood types and components. RBC and TC generally have higher standard deviation, skewness, and kurtosis compared to FFP, indicating more irregular distributions. For example, RBC in type A+ shows the highest variability with standard deviation = 24.234 and extreme kurtosis = 40.058, with a maximum value of 315, far exceeding its mean of 7.750. In contrast, FFP demand across all blood types generally has lower skewness and kurtosis, suggesting more stable demand patterns.

Forecasting Models

AutoRegressive Integrated Moving Average (ARIMA)

ARIMA is a traditional time series forecasting model that remains widely used due to its simplicity and interpretability. It performs well when dealing with data that exhibit clear trends and seasonality, but it faces challenge when dealing with highly non-linear patterns, requires stationary data, and more suitable for short to medium-term forecasting than the long-term.[22]

Figure 1. ARIMA flow diagram



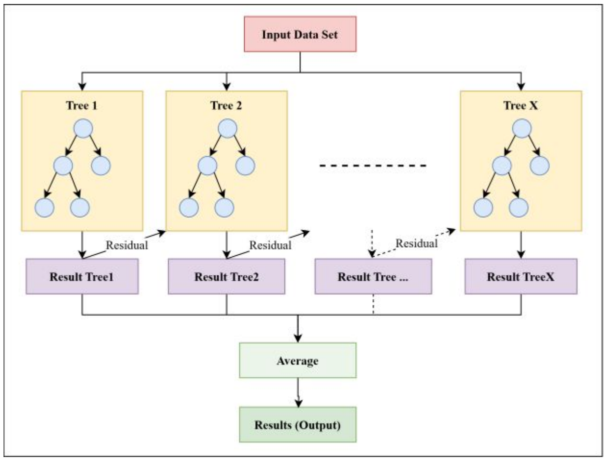
As shown in Figure 1, the ARIMA modeling process begins with a stationarity test. If the time series is found to be non-stationary, differencing (I) is applied to eliminate trends and seasonality. The Auto Regressive (AR) component models the relationship between past and present values using lagged observations, which refers to previous values in the time series. The Moving Average (MA) component improves accuracy by considering past forecasting errors. ACF analysis is used to determine the Moving Average (MA) parameter (q), while PACF analysis helps in determine the number of lags in Auto Regressive (AR) parameter (p). ARIMA selects the best combination of AR, I, MA components (commonly referred as p, d, q parameters) based on AIC/BIC criteria. Once trained, the model predicts future values based on past values.

ARIMA can handle linear patterns effectively due to its components. AR captures linear dependencies, I ensures stationarity, and MA reduces short-term noise. However, it struggles with non-linear relationships and long-term dependencies. Alternative models, such as XGBoost, NN, LSTM, and TCN more suitable for capturing non-linear relationships.

Extreme Gradient Boosting (XGBoost)

XGBoost is a machine learning algorithm based on gradient boosting decision trees, which iteratively learn from previous errors to improve performance. It is scalable and capable of capturing complex non-linear relationships, making it superior to traditional statistical models in many forecasting tasks.[23]

Figure 2. XGBoost structure



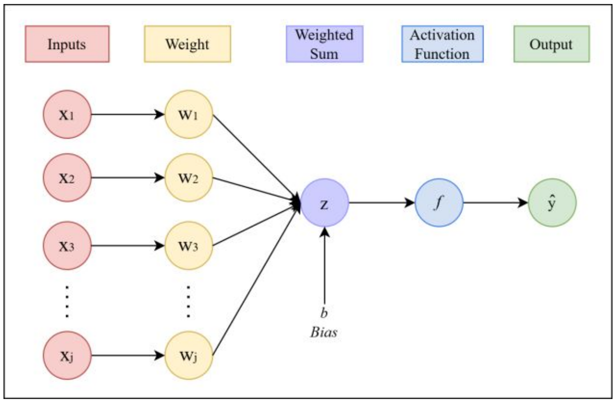
As illustrated in Figure 2, XGBoost starts by training an initial decision tree as a weak learner. The residual errors from this first tree are then used to train the next tree in an iterative process. This approach enables sequential learning, allowing each tree focus on correcting the of previous one, ultimately leading to optimized results. To improve generalization and prevent overfitting, XGBoost applies L1/L2 regularization, L1 (Lasso Regression) helps in feature selection and L2 (Ridge Regression) makes model more stable. For better efficiency, XGBoost leverages parallel execution and pruning techniques by removing the unnecessary branch making it much faster in processing the model. In the final step, the ensemble of trees is combined to generate prediction.

XGBoost excels in handling large datasets and complex relationships, making it a powerful alternative to ARIMA. However, since it does not inherently capture temporal dependencies, additional feature engineering (e.g., lag features) is required for time series forecasting. While effective, it may not be as robust as sequential models like LSTM or TCN in capturing long-term dependencies.

Neural Networks (NN)

Based on structure, neural networks have flexibility and capability in complex functions, making them effective for capturing non-linear relationships in time series data. The simplest form, the perceptron neural network, consists of input neurons, weighted connections, and activation functions.

Figure 3. Perceptron neural network architecture



As shown in Figure 3, the input layer receives values as input features (x_j). The model then applies pre-initialized weights (w_j) and biases (b) to adjust the decision boundary during training. The weighted sum calculation determines how strongly inputs influence the output, as presented in Equation 1.

$$z = \sum_{j=1}^{n} w_j x_j + b \tag{1}$$

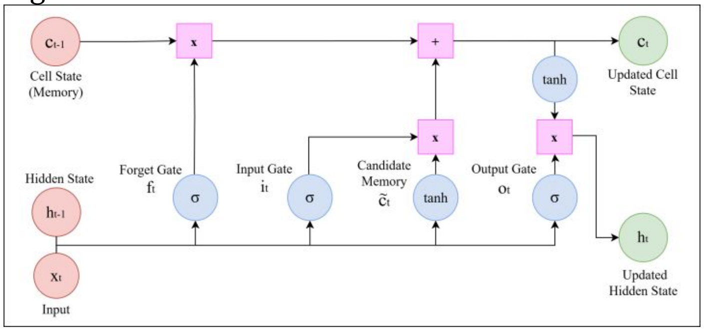
This value is then passed through an activation function (f), which allows the model to capture non-linear relationships. When receiving input, the neuron applies this activation function to the signal, introducing nonlinearity to the model.[24] Unlike traditional statistical models, neural networks can process information from multiple perspectives by utilizing different types of neurons, including feature extraction neurons, computational neurons (which may undergo dropout for regularization), and output neurons that generate final predictions ($\hat{y}$). The learning rate controls weight adjustments, leading to an iterative weight update process until convergence.

Neural networks are effective for pattern recognition due to their ability to process information from multiple perspectives. However, they process each input independently and do not retain past information or learn from previous time steps, making it unable capture time dependencies in sequential data. They also require a large amount of data for accurate forecasting, need more computational resources, and may tend to overfit if they not properly used.[25] In time series forecasting, more specialized architectures like LSTM and TCN are preferable, as they are specifically designed to model sequential patterns and long-term dependencies.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber in 1997, is a type of Recurrent Neural Network (RNN) designed to overcome the "vanishing gradient" problem, allowing it to determine which information should be retained and which should be discarded.[16]

Figure 4. LSTM architecture



As depicted in Figure 4, LSTM is designed to capture long-term dependencies in sequential data using three gates including forget gate, input gate, and output gate. The forget gate determines which information from previous cell state (c_{t-1}) should be discarded. This is achieved by applying a sigmoid activation function (σ) to the previous hidden state (h_{t-1}) and current input (x_t), generating a forget gate output (f_t) that decides the proportion of information to retain or forget. The relevant information is then scaled (×) through element wise multiplication with the previous cell state (c_{t-1}).

The input gate determines which new information should be added to the cell state. It processes the previous hidden state (h_{t-1}) and current input (x_t) using a sigmoid function to generate an input gate output (i_t). Additionally, a candidate memory ($\tilde{c}_t$) is produced using tanh activation function, allowing the LSTM to preserve both positive and negative signals. Before updating the cell state, the element wise multiplication (×) of the input gate and candidate memory ensures that only the most relevant information is added. The updated cell state (c_t) is then calculated by adding (+) the scaled previous cell state and the scaled candidate memory, ensuring relevant patterns or information are retained over time.

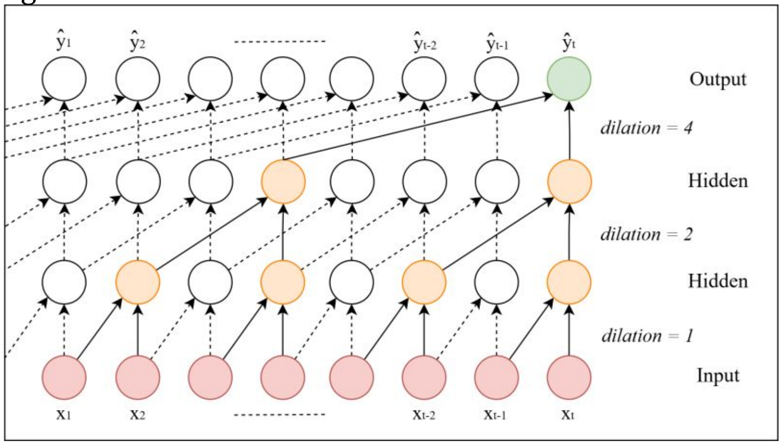
The output gate determines the useful information to pass to the next hidden state (h_t) and final output. This gate applies a sigmoid function (σ) to the previous hidden state (h_{t-1}) and current input (x_t) to generate the output gate value (o_t). The updated cell state (c_t) is carried forward to next time step, to store long term dependencies. Meanwhile, the final hidden state (h_t) is computed by applying tanh activation function to the updated cell state (c_t) and then performing element wise multiplication (×) with the output gate value (o_t). This hidden state serves as the output of each time step and is passed to the next time step along with the cell state (c_t), which continuously preserves relevant information over time.

LSTM outperforms simpler models like ARIMA in dealing with long-term dependencies. It has been widely applied in forecasting applications including forecast and recognition of text and sound.[26] However, as a deep learning model, it requires a substantial amount of data to achieve reliable results. Additionally, its sequential nature makes training computationally expensive and limits parallelization. Compared to TCN, LSTM is challenging to optimize due to its complex gating mechanism and higher parameter count. The gating mechanism, forget, input, and output, which regulate the flow of information through the network. Each gate has its own set of weights and biases, increasing the number of trainable parameters. While LSTM remains strong choice for sequence modeling, TCN offers more efficient alternative by enabling parallel computation and capturing long-term dependencies through dilated convolutions.

Temporal Convolutional Network (TCN)

TCN is an alternative to LSTM that leverages convolutional layers for sequence modeling. In several studies, this model outperforms RNN.[27] TCN capture the short and long-term dependencies through dilation and causal convolution. As the strength of its nature, TCN capable to handle variable-length sequences naturally, making them versatile for various time-series applications.[27]

Figure 5. TCN architecture



As depicted in Figure 5, TCN process sequential data by passing the input sequence x_1, x_2, …, x_t through multiple convolutional layers. Unlike recurrent models, TCN employ causal convolutions, ensuring that each time step xt only influences the present and future outputs, effectively preventing information leakage from future time steps.

Each convolutional layer applies dilated convolutions to expand the receptive field. By progressively increasing the dilation rates (e.g., 1, 2, 4, 8), the network capture patterns over varying temporal scales. This architecture allows the model to process long sequences efficiently by learning dependencies across both short and long-time horizons. The convolutional layers transform the input sequence into hidden representations through a series of operations that maintain the hierarchical structure of the data.

The outputs $\hat{y}_1$, $\hat{y}_2$, ... $\hat{y}_t$ are generated once the processed information reaches the output layer. These outputs maintain the sequence structure and dependencies identified throughout the network. As a result, the outputs reflect the patterns and relationships learned from the original input sequence.

TCN is particularly efficient because it allows for parallel computation, unlike LSTMs that process sequence sequentially. However, in certain cases, LSTM still provides better accuracy, especially when sequential dependencies are highly complex. While TCN has shown promising results, it is less widely adopted in forecasting compared to LSTM. This paper aims to evaluate its performance in real-world applications to assess its effectiveness in capturing temporal dependencies.

Hyperparameter Selection

Hyperparameters for each model were tuned to achieve their best performance. Each model applied different tuning approaches, adjusted to the characteristics of its respective algorithm.

AutoRegressive Integrated Moving Average (ARIMA)

To optimize the model, Auto-ARIMA was used to identify the optimal set of parameters (p, d, q) and seasonal parameters (P, D, Q, m). The search started from zero for all orders and was constrained with max_p and max_q set to 5 to prevent overfitting. Weekly seasonality was incorporated by setting $m = 7$ and enabling the seasonal parameter. The non-seasonal differencing order d was fixed at 0, while seasonal differencing D was set to 1. The model selection was based on the lowest AIC score, with the stepwise search option enabled to reduce computation time.

Extreme Gradient Boosting (XGBoost)

The model was trained using a sliding window of the previous 7 days as input features. Hyperparameters were kept close to the defaults, with $n\_estimators$ set to 100, learning rate fixed at 0.1, and random state at 42 for reproducibility. No extensive tuning was performed, as the primary goal was to benchmark against deep learning models rather than to optimize XGBoost specifically. All input features were scaled using MinMaxScaler to maintain consistency in feature ranges.

Neural Networks (NN)

The model was tuned into two stages. In the first stage, RandomizedSearchCV was applied to identify optimal hidden layer sizes, exploring combinations in the range of 10 to 150 neurons with a step of 20. In the second stage, Optuna was used to fine-tune the alpha (L2 regularization term) within the range 0.1–1.0 and the $learning\_rate\_init$ between 0.001 and 0.1. The optimization objective was to maximize the validation R^2 score. This sequential approach allowed the architecture and learning rate parameters to be optimized independently for better efficiency.

Long Short-Term Memory (LSTM)

The LSTM model was tuned using KerasTuner with Random Search method. The search was employed to explore a range of hyperparameters, including the number of units (32 to 256 in steps of 32), dropout rate (0.0 to 0.5), and learning rate (1×10^{-4} *to* 1×10^{-2} on a logarithmic scale). The tuning process was configured with 30 trials and two executions per trial, with each trial trained for 20 epochs. Early stopping was applied based on validation loss to avoid overfitting. The final configuration was chosen based on the lowest MSE.

Temporal Convolutional Network (TCN)

The TCN model was also tuned using KerasTuner Random Search, exploring the number of convolutional filters (32 to 256), kernel size (2 to 8), number of dilation levels (1 to 4), skip connection activation (True = enabled/False = disabled), dropout rate (0.0 to 0.5), and learning rate (1×10^{-4} *to* 1×10^{-2}). Similar to the LSTM tuning process, 30 trials with two executions per trial were run, and the best configuration was selected based on validation performance.

Model Performance Evaluation

Evaluation of forecasting results can be seen from the difference between the estimates and actual data (forecast vs actual). Parameters that can be seen to measure the accuracy of forecasting results is to look at the level of error. The smaller the resulting error value indicates a high level of forecasting accuracy. In this study, MAE and RMSE are used as evaluation parameters for the models.

Mean Absolute Error (MAE)

MAE is a commonly used metric for evaluating model performance. It measures the average absolute difference between predicted and actual values (Equation 2),

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

where n is total number observations, y_i represents actual value and $\hat{y}_i$ is predicted value of i-th observation. The absolute error, $|y_i - \hat{y}_i|$, ensures that all error is treated equally, preventing positive and negative errors from cancelling each other out.[28] It aggregates error across all observations, providing a comprehensive measure of model performance. This total is then divided by the total number of observations (n), making MAE easy to interpret as the typical deviation of predictions from actual values.

MAE is particularly useful when the primary objective is to measure the typical magnitude of errors, as it maintains the same unit as the target variable. Additionally, MAE is less sensitive to outliers, as all errors contribute proportionally to the metric without being squared. However, its uniform treatment of errors means that large deviations are not penalized more than small ones, which can be a limitation in scenarios where large forecasting errors have a significant impact.

Root Mean Square Error (RMSE)

RMSE is an alternative metric that places a greater emphasis on large errors by squaring the residuals before averaging them. The RMSE formula is presented in Equation 3.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad (3)$$

Similar to MAE where n is total number observations, y_i and $\hat{y}_i$ are actual and predicted value of i-th observation, respectively. Unlike MAE, RMSE squares each error term before averaging, $(y_i - \hat{y}_i)^2$, ensures that all values remain in positive form and amplifies larger errors, making RMSE more sensitive to significant deviations. The square root then restores the unit to match the original target variable, making RMSE easier to interpret.[28]

RMSE is particularly useful in scenarios where large deviations are undesirable, such as safety critical predictions. This is because RMSE applies greater penalties to larger errors by squaring each error before averaging, making it more sensitive to significant deviations. However, this sensitivity to large error also makes RMSE more susceptible to outliers, which may distort performance evaluations if extreme values are present.

Table 2. Forecasting model performance results

Blood Component	Model	A+		B+		AB+		O+		Standard Deviation		Average	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Fresh Frozen Plasma (FFP)													
	ARIMA	4.078	6.017	8.058	9.385	4.918	6.365	7.422	11.348	1.921	2.544	6.119	8.279
	XGBoost	4.470	6.530	5.260	8.320	3.150	5.020	6.360	9.370	1.351	1.925	4.810	7.310
	NN	3.456	4.715	4.683	7.121	2.419	4.341	14.458	21.124	5.547	7.962	6.254	9.325
	LSTM	4.420	6.177	5.551	8.176	4.494	6.493	5.721	9.168	0.685	1.414	5.047	7.504
	TCN	3.436	5.040	4.844	7.908	4.468	6.880	5.631	9.237	0.912	1.770	4.594	7.266
Red Blood Cells (RBC)													
	ARIMA	17.663	23.004	35.697	39.482	8.172	10.154	20.865	26.369	11.417	12.052	20.599	24.752
	XGBoost	15.670	19.620	17.780	21.990	6.820	8.440	18.360	22.660	5.351	6.621	14.658	18.178
	NN	15.320	19.154	12.907	16.612	7.714	9.289	18.487	22.556	4.545	5.630	13.607	16.903
	LSTM	13.037	16.712	13.447	16.591	6.211	7.303	16.686	20.280	4.403	5.549	12.345	15.221
	TCN	15.322	20.016	13.248	16.706	6.295	7.503	17.290	21.796	4.789	6.361	13.039	16.505
Thrombocyte Concentrate (TC)													
	ARIMA	9.694	12.334	9.507	12.107	5.559	7.555	16.931	22.160	4.739	6.154	10.423	13.539
	XGBoost	7.410	9.960	8.420	10.870	5.620	7.550	13.280	17.010	3.276	4.026	8.683	11.348
	NN	6.281	8.261	7.889	9.679	4.485	6.780	9.344	12.477	2.092	2.427	7.000	9.300
	LSTM	6.498	8.651	7.189	9.268	4.645	6.607	9.118	11.585	1.848	2.049	6.863	9.028
	TCN	6.283	8.118	7.975	9.896	5.101	7.267	9.281	12.216	1.842	2.188	7.160	9.374

Results

Overall Performance Results

After applying each forecasting model to the historical blood demand data, the results were analyzed across different blood components. Table 1 summarizes the performance of each model using MAE and RMSE as evaluation metrics. To facilitate interpretation, the standard deviation and average of MAE and RMSE across the four blood types (A+, B+, AB+, and O+) are also provided. The "yellow cell" indicates the model with the lowest standard deviation and average of MAE and RMSE.

As shown in Table 2, for the FFP blood component, the TCN model demonstrated superior performance across most blood types. Specifically, for A+, TCN achieves the lowest error rates (MAE = 3.436, RMSE = 5.040). NN records a slightly lower RMSE of 4.715. For B+, NN yields the best performance with MAE = 4.683 and RMSE = 7.121. These scores are followed by TCN with MAE = 4.844 and RMSE = 7.908. For AB+, NN performs best again with MAE = 2.419 and RMSE = 4.341. For O+, TCN has the lowest MAE score with 5.631 and LSTM has the lowest RMSE score with 9.168.

Among all models, the LSTM model achieved the lowest standard deviation (MAE = 0.685, RMSE = 1.414), indicating greater stability in handling errors for FFP components. Additionally, the TCN model outperformed the other models on average, as reflected by its lowest average of MAE = 4.594 and RMSE = 7.266. In contrast, the NN model exhibited the highest standard deviation and average of MAE and RMSE, suggesting its forecasting performance was inconsistent, particularly evident in the highest MAE and RMSE scores observed for blood type O+.

For the RBC blood component, the LSTM model consistently demonstrated the best performance, as indicated by the lowest MAE and RMSE scores across all blood types. Among all the lowest MAE scores, only for blood type B+ did the NN model achieve the best result (MAE = 12.907), followed closely by the LSTM model (MAE = 13.248). These results are further supported by the standard deviation and average of MAE and RMSE, in which the LSTM model obtained the lowest scores. Meanwhile, traditional statistical methods such as ARIMA showed the weakest performance, with a standard deviation of MAE = 11.417 and RMSE = 12.052, and an average of MAE = 20.599 and RMSE = 24.752.

For the TC blood component, the LSTM and TCN models outperformed other models. For A+, TCN showed the lowest RMSE = 8.118 and NN achieved the lowest MAE = 6.281. For B+, LSTM achieved the lowest error with MAE = 7.189 and RMSE = 9.268. For AB+, it exhibited NN as the lowest MAE = 4.485 and LSTM with the lowest RMSE = 6.607. For O+, LSTM has the lowest MAE (5.393) and RMSE (11.585).

The lowest standard deviation for the TC blood component was achieved by LSTM (MAE = 1.848, RMSE = 2.049) and TCN (MAE = 1.842, RMSE = 2.188). This suggests that LSTM and TCN models consistently performed well across all blood types. Overall, the TC blood component showed the lowest average of MAE and RMSE, achieved by LSTM. Similar to the other components, particularly RBC, ARIMA showed the weakest performance among all models, as indicated by its high standard deviation and average scores of MAE and RMSE.

While this study evaluates all blood types for various blood components (Fresh Frozen Plasma or FFP, Red Blood Cells or RBC, and Thrombocyte Concentrate or TC), the subsequent analysis and illustrations primarily focus on blood type O+. This approach aims to streamline the explanation while maintaining relevance, given that blood type O is often the most sought after and commonly utilized in transfusion procedures (Simpson, 2020). Focusing the analysis on blood type O+ helps illustrate the model's performance. The trends and patterns identified for blood type O+ are intended to provide insights that could be applicable to other blood types, allowing for a more streamlined interpretation throughout the study.

Figure 6. Fresh Frozen Plasma (FFP) forecast for type O

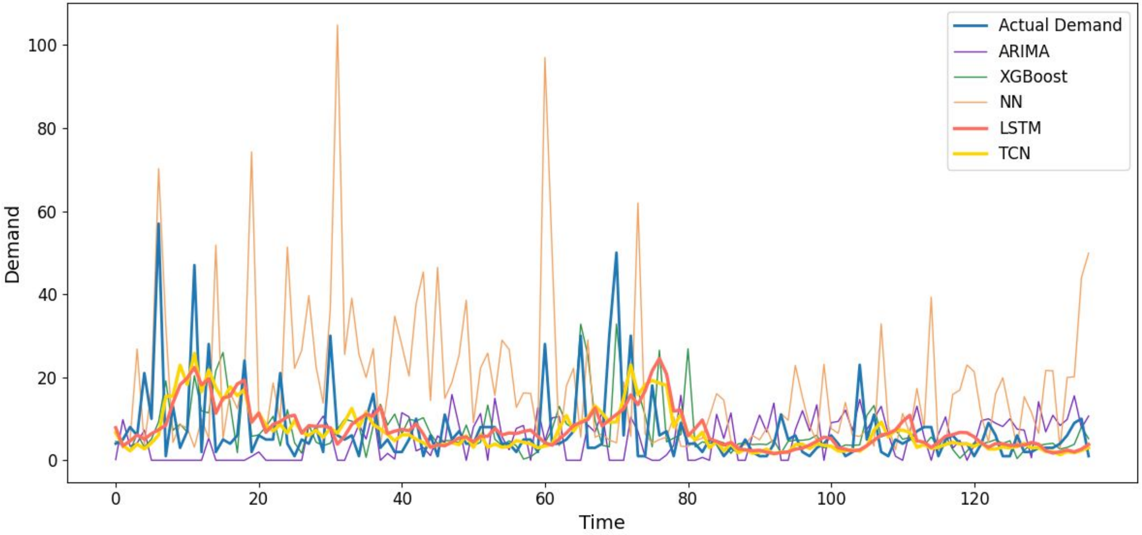


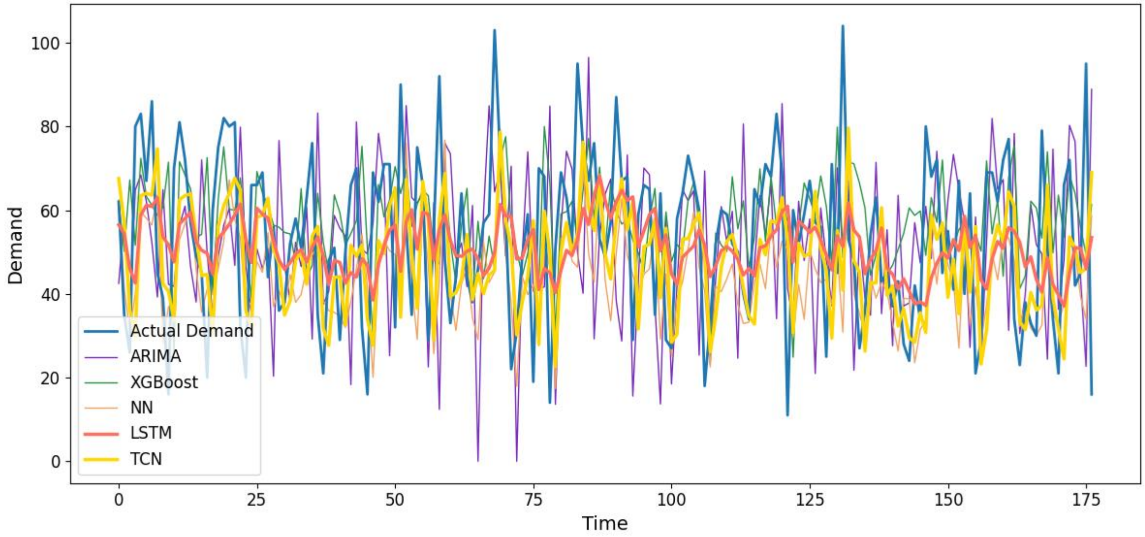
Figure 6 presents the forecasting results for the FFP blood component, demonstrating that deep learning models generally perform well in capturing the overall trend. The orange and yellow lines represent the LSTM and TCN forecasting results, respectively. The use of dilated convolutions enables TCN to effectively recognize hierarchical patterns over longer time spans. The LSTM model follows with slightly different forecasting behavior compared to TCN, further supporting the observation that deep learning approaches tend to outperform traditional methods.

However, while both LSTM and TCN excel at identifying general trends, they struggle to accurately capture sudden spikes in demand. Their forecasted curves appear overly smooth and are often unable to react swiftly to abrupt changes, leading to underestimations during peak periods.

In addition, Figure 6 shows that XGBoost (green line) produces a forecast that closely follows the actual demand. This is consistent with its relatively low MAE and RMSE values (Table 2), ranking just behind TCN and LSTM. Nevertheless, it occasionally fails to capture certain demand points, such as at index Time = 80.

In contrast, ARIMA (purple line) frequently underestimates the actual demand, predicting consistently lower values. Similarly, the Neural Network (light orange line) performs poorly, producing forecasts that significantly deviate from the actual demand curve. Therefore, these two models show limited effectiveness in forecasting blood type O for the FFP blood component.

Figure 7. Red Blood Cells (RBC) forecast for type O



As depicted in Figure 7, the LSTM model demonstrates effective forecast performance for RBC demand by capturing the overall trend and maintaining stability in its predictions. Its forecast aligns reasonably well with the actual demand, particularly during periods of moderate fluctuations. Similarly, the TCN model exhibits strong performance, closely following the actual demand pattern of the RBC blood component. However, LSTM tends to smooth out abrupt variations, which results in reduced responsiveness to sudden spikes in demand.

XGBoost and the Neural Network (NN) perform moderately well, producing reasonably accurate forecasts that generally align with the actual demand. However, their performance remains inferior to that of the LSTM and TCN models. ARIMA, on the other hand, delivers the poorest performance among all models, showing large deviations and failing to capture the actual demand trend.

Figure 8. Thrombocyte Concentrate (TC) forecast for type O

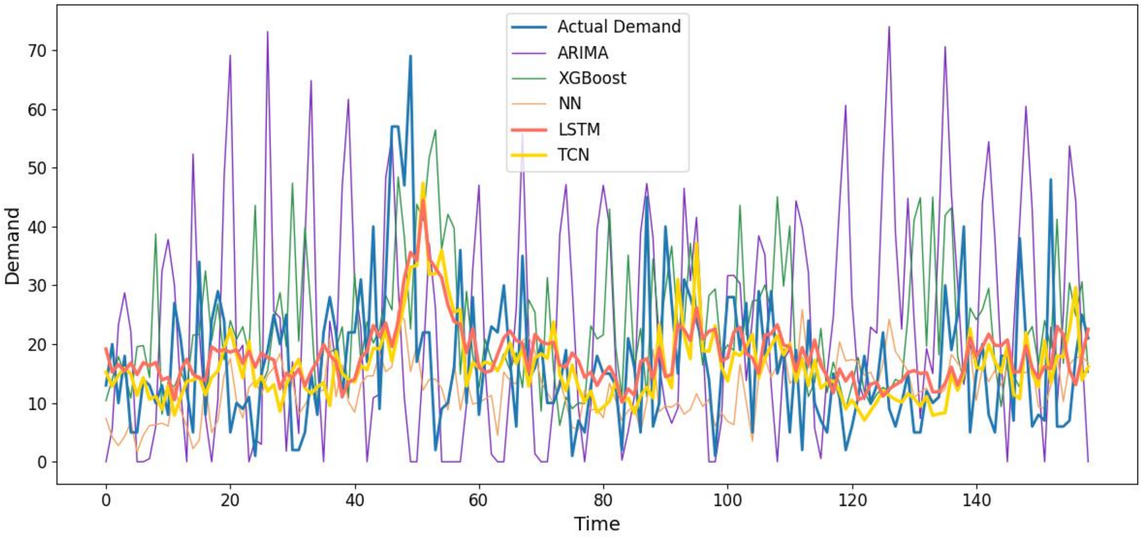


Figure 8 presents the forecasting results for TC demand. The TCN and LSTM models effectively capture the overall trend, demonstrating their ability to follow broader fluctuations over time. However, both models exhibit limitations in accurately predicting sudden spikes in demand. Similar to the results observed for FFP and RBC, the forecasted curves tend to appear overly smooth and less responsive to abrupt, isolated peaks.

The Neural Network (NN) also demonstrates good performance, although it occasionally fails to accurately capture the actual demand, placing it behind other deep learning models. In contrast, the ARIMA and XGBoost models exhibit large deviations from the actual demand curve, suggesting that they are less suitable for forecasting the TC blood component for blood type O.

Discussion

Principal Results

This study evaluated the forecasting performance of two deep learning model, Temporal Convolutional Network (TCN) and Long Short-Term Memory (LSTM), in predicting demand for Fresh Frozen Plasma (FFP), Red Blood Cells (RBC), and Thrombocyte Concentrate (TC). The results indicate that both TCN and LSTM consistently achieved lower error metrics (MAE and RMSE) compared to traditional models such as ARIMA, XGBoost, and Neural Networks (NN).

Table 2 presents the standard deviation of each model's performance, showing that LSTM and TCN consistently outperform traditional models. This suggests that deep learning models offer greater stability in forecasting the three blood components. Among them, LSTM demonstrates superior stability in forecasting FFP and RBC demand, while for TC demand, both TCN and LSTM perform equally well.

Specifically, TCN yields the lowest standard deviation in MAE, while LSTM produces the lowest RMSE deviation.

In terms of average MAE and RMSE scores, TCN and LSTM also outperform traditional models. TCN performs particularly well for FFP, whereas LSTM excels in forecasting RBC and TC demand. For FFP, TCN achieved a MAE of 4.594 and RMSE of 7.266. In RBC forecasting, LSTM recorded a MAE of 12.345 and RMSE of 15.221. For TC, LSTM obtained a MAE of 6.863 and RMSE of 9.028. These results are further supported by visual comparisons in Figures 6, 7, and 8 (Appendix).

Deep learning models, particularly LSTM and TCN, are effective due to their ability to learn both short- and long-term patterns. LSTM, with its sequential processing and memory architecture, is well-suited for capturing temporal dependencies. Meanwhile, TCN leverages dilated convolutions, allowing it to recognize temporal patterns over broader contexts. Although TCN and LSTM often produce similar outcomes, some differences are observed. For example, in forecasting blood type O in the FFP component, TCN achieved the lowest MAE (5.631), while LSTM obtained the lowest RMSE for the same case. Despite LSTM's strong RMSE performance, some forecasted points appear less accurate during sudden spikes. This is likely due to LSTM's sequential nature, where data is processed step-by-step, limiting its ability to quickly adapt to abrupt changes.

Across all blood components and types, deep learning models in this study, LSTM and TCN, consistently outperform traditional statistical and machine learning approaches. Their strength lies in modeling sequential dependencies, nonlinear patterns, and seasonal fluctuations inherent in blood demand data. LSTM is particularly effective for highly volatile series due to its capacity to capture long-term dependencies. On the other hand, TCN offers robust temporal pattern recognition while maintaining computational efficiency.

Although ARIMA, XGBoost, and NN occasionally yield relatively low error scores, their model architectures are not inherently designed to capture temporal dependencies as effectively as sequence-based models like TCN and LSTM. ARIMA, with its linear assumptions, struggles with nonlinear and long-range dependencies, resulting in higher error values. XGBoost, as a tree-based ensemble method, performs moderately well but lacks temporal awareness unless time-based features are manually engineered. Similarly, a standard feedforward Neural Network (NN) does not incorporate temporal context unless extended with recurrent or convolutional layers. Nevertheless, NN demonstrates reasonable performance under simpler, more stable demand patterns.

Limitations

While this study emphasizes maintaining strong forecasting performance with low standard deviation as an indicator of model stability, it also has several limitations.

First, the forecasting models demonstrated difficulty in detecting sudden and irregular fluctuations in blood demand. Both LSTM and TCN are designed to learn general and recurring temporal patterns, which may limit their responsiveness to abrupt variations. This poses a challenge in real-world applications, where emergencies, seasonal shifts, or special events can cause unexpected demand surges.

Specifically, LSTM tends to smooth out sharp variations, reducing its ability to respond accurately to sudden spikes. This limitation stems from its sequential processing architecture, where predictions are based heavily on prior time steps. Although this makes LSTM effective for learning consistent temporal dependencies, rapid changes in data may be diluted or averaged out across the sequence, resulting in missed peaks or drops.

Similarly, TCN also struggles to capture abrupt changes in demand. Its forecasted output often appears overly smooth and fail to adapt quickly during peak periods. This is primarily due to the use of dilated convolutions, which are optimized for learning long-term dependencies by skipping over certain data points. While effective for broad trend recognition, this architecture may overlook sharp, isolated fluctuations leading to underestimations during sudden demand spikes.

In addition, the descriptive statistic (Table 1) indicates that several blood type and component combinations have strongly skewed distributions with high kurtosis, suggesting that demand is usually low but occasionally spikes to very high levels. For example, RBC demand in A+ and O+ can reach values many times higher than average. These rare surges are important from a clinical perspective, but their infrequency in the historical record makes them difficult for the models to capture. Consequently, both LSTM and TCN often underestimate demand during such peak events, which contributes to their limited responsiveness.

Second, the dataset used in this study is limited in terms of geographic and contextual diversity, which may restrict the generalizability of the findings. Blood usage patterns can vary significantly across regions, healthcare institutions, and population demographics. Therefore, the performance of the models may differ in other settings not represented in the current dataset. However, no evidence of systematic bias was identified in the data collection process, as the records were obtained directly from standardized operational reporting of the Indonesian Red Cross.

Comparison with Prior Work

This study builds upon previous research efforts that primarily employed traditional time series and classical machine learning techniques for blood demand forecasting. For example, Shih and Rajendran (2019) reported that ARIMA delivered the lowest forecasting error among conventional statistical models when applied to blood component demand. Similarly, Fanoodi et al. (2019) examined both ARIMA and Artificial Neural Networks (ANN) for daily blood demand prediction. Sun et al. (2021) advanced the field by employing XGBoost, a tree-based ensemble model, to forecast red blood cell demand. Their results demonstrated enhanced pattern recognition over older models and showed promise in handling moderately complex demand structures.

However, while these approaches brought incremental improvements, they remained limited in their ability to fully capture complex temporal dynamics, particularly in the presence of long-term dependencies, seasonality, and sudden, irregular demand spikes. Traditional models like ARIMA rely on linear assumptions and are effective only for stationary series with stable trends. Meanwhile, machine learning models such as XGBoost and ANN lack intrinsic mechanisms to understand the sequential nature of time series data unless temporal features are explicitly engineered.

In contrast, this study explores the use of deep learning models, specifically Temporal Convolutional Networks (TCN) and Long Short-Term Memory (LSTM), which are inherently designed to model temporal structure, sequential dependencies, and nonlinear relationships in time series data. TCN, with its dilated convolutional layers, can capture long-range patterns while remaining computationally efficient. LSTM, with its gated memory structure, effectively retains historical context over extended sequences.

Our empirical findings show that TCN and LSTM consistently outperform ARIMA, ANN, and XGBoost across multiple blood components (FFP, RBC, and TC) and blood type (A+, B+, AB+, and O+). Notably, models with a built-in understanding of temporal structure, such as TCN and LSTM exhibited greater stability, lower average error, and better adaptability to trend variations, especially in datasets with nonstationary and fluctuating patterns.

Furthermore, the advantage of deep learning becomes more pronounced in multi-step and medium- to long-range forecasting, where capturing seasonal cycles and gradual demand shifts is crucial. Although models like XGBoost are competitive in forecasting under relatively stable conditions, their performance deteriorates when applied to more volatile or irregular patterns, scenarios where TCN and LSTM maintain robustness.

Despite these advancements, it is important to acknowledge that even TCN and LSTM face challenges in forecasting extreme values or rare spikes. This is a common limitation in many forecasting models and highlights an area for future research, potentially involving hybrid models that leverage its respective strengths (integrating LSTM and TCN models).

Conclusions

This study has demonstrated that Temporal Convolutional Networks (TCN) and Long Short-Term Memory (LSTM) outperform traditional forecasting methods such as ARIMA, XGBoost, and standard Neural Networks in predicting demand for Fresh Frozen Plasma (FFP), Red Blood Cells (RBC), and Thrombocyte Concentrate (TC). These models consistently achieved lower error metrics (MAE and RMSE) and demonstrated greater stability, indicating their effectiveness in modeling the complex temporal dynamics inherent in blood demand data.

The comparative advantage of these models lies in their distinct strengths. TCN offers efficient learning of broad temporal patterns through dilated convolutions, while LSTM excels at capturing long-term sequential dependencies through its gated memory architecture. Both models are particularly well-suited for recognizing recurring patterns and trends, which are essential in time series forecasting for healthcare logistics. However, their performance diminishes in scenarios involving sudden, unpredictable demand spikes, highlighting the need for further improvement in responsiveness.

Based on these findings, TCN and LSTM are recommended as reliable tools for blood demand forecasting, especially when used in operational environments where accurate forecasting is essential. The integration of these two architectures, leveraging TCN's parallelism and LSTM's memory depth, may yield even more robust forecasting systems capable of balancing accuracy and adaptability.

Future work should consider enhancing these models through integrating these models. Additionally, expanding the dataset to encompass multiple geographic regions and healthcare institutions would improve model generalizability and support the development of more universally applicable forecasting systems.

Acknowledgements

This research was supported by Seed Funding Hibah Professor, Lektor Kepala, dan Lektor, Nomor: NKB-3442/UN2.F4.D/PPM.00.00/2024. All authors contributed significantly to the conception, methodology, analysis, and writing of the manuscript. EL led the project design, data analysis, model development, and manuscript preparation. FA contributed to the interpretation of results and manuscript writing. ZZ contributed to model development, interpretation of results, and critical revision of the manuscript. All authors reviewed and approved the final version of the manuscript.

Conflicts of Interest

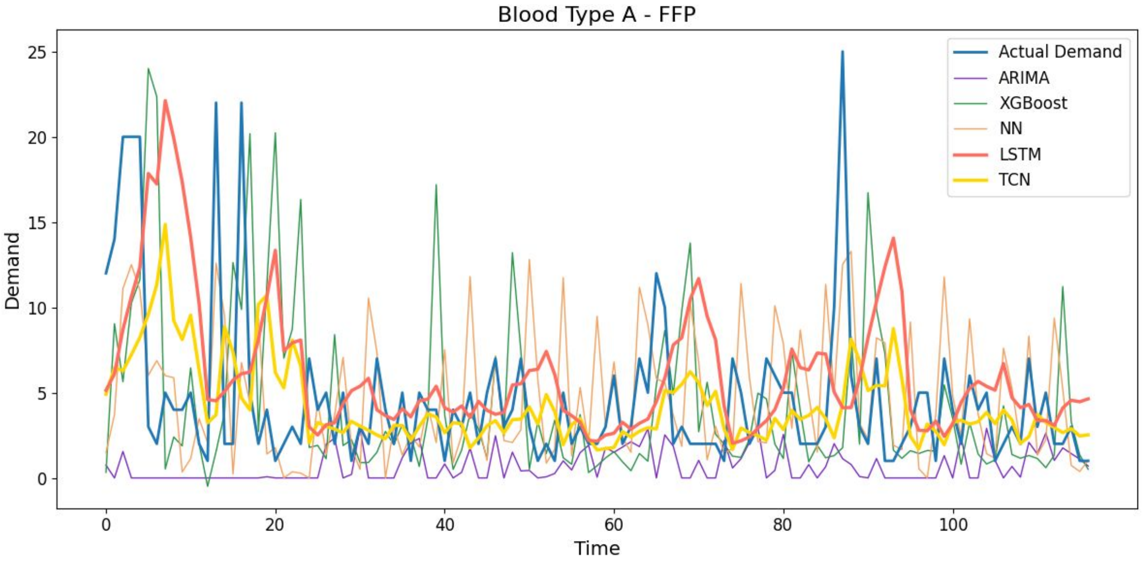
None declared.

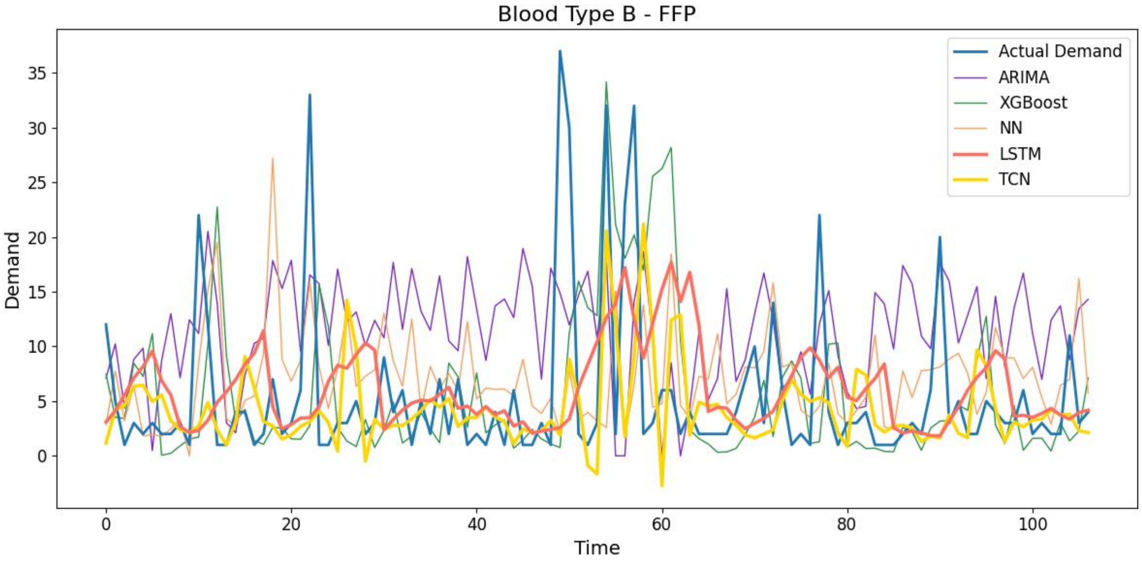
Abbreviations

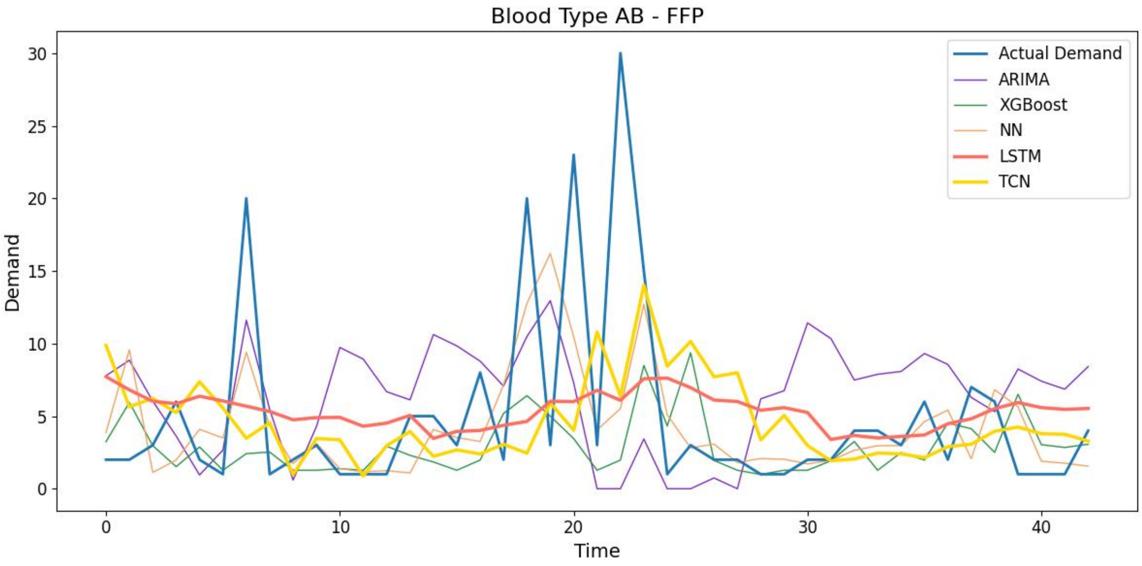
ARIMA: autoregressive integrated moving average
FFP: fresh frozen plasma

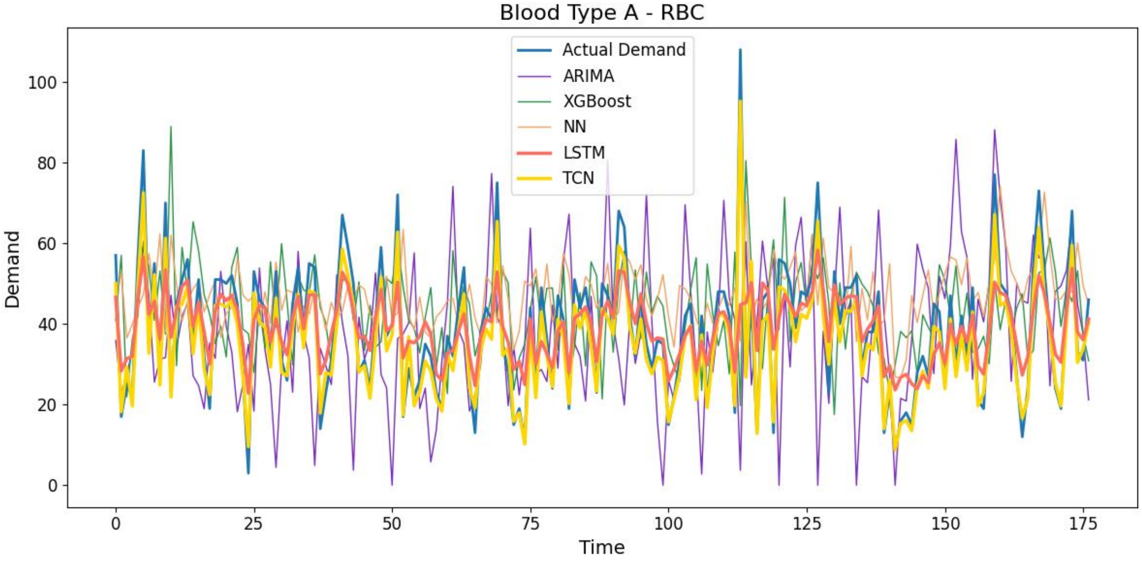
LSTM: long short-term memory
MAE: mean absolute error
MSE: mean square error
NN: neural network
RBC: red blood cells
RMSE: root mean square error
TC: thrombocyte concentrate
TCN: temporal convolutional network
XGBoost: extreme gradient boosting

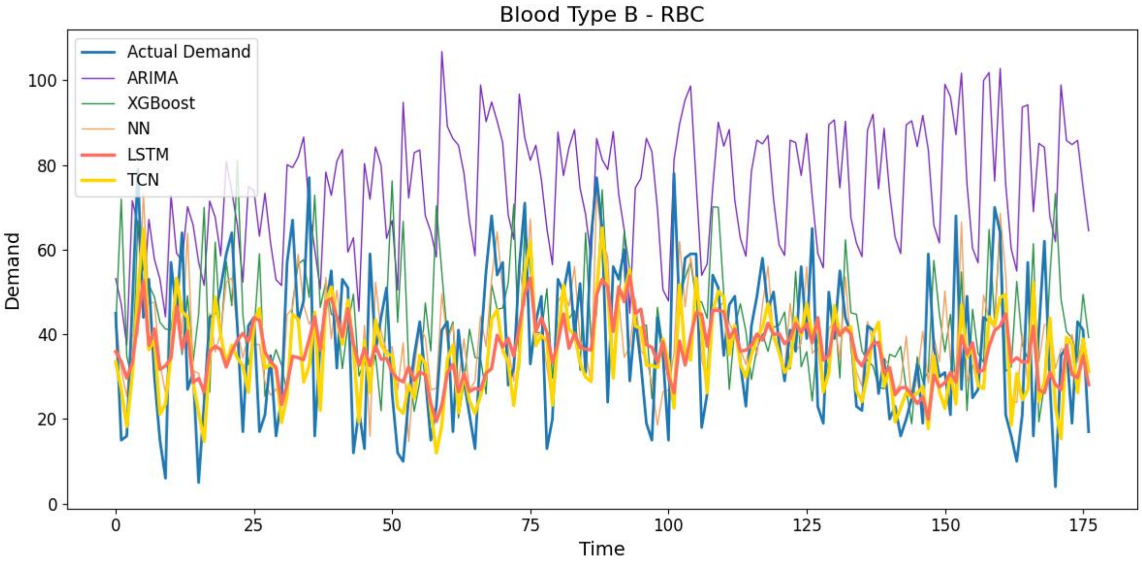
Appendix

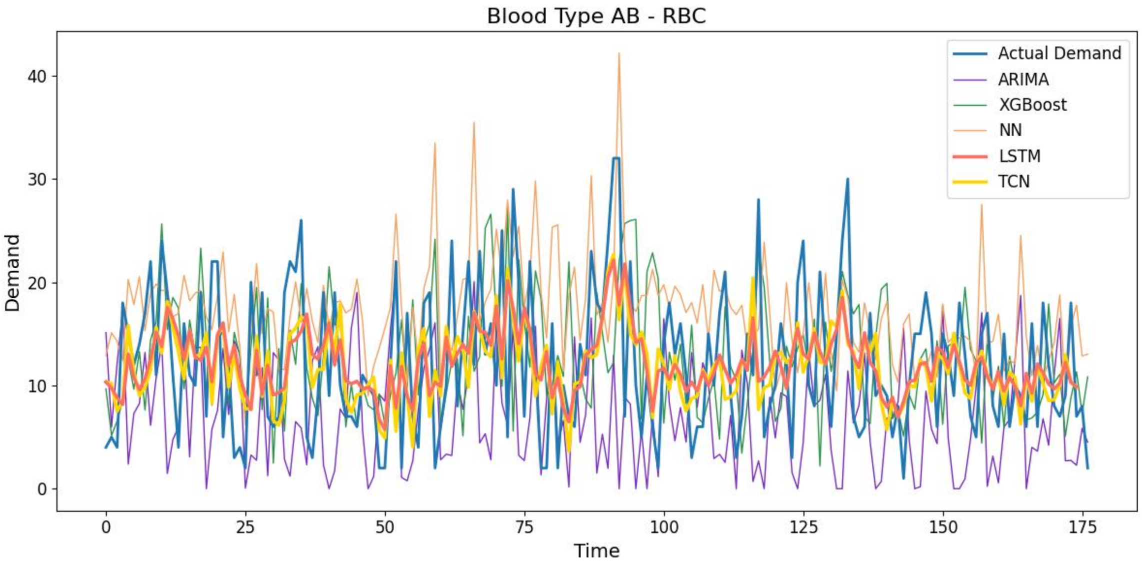


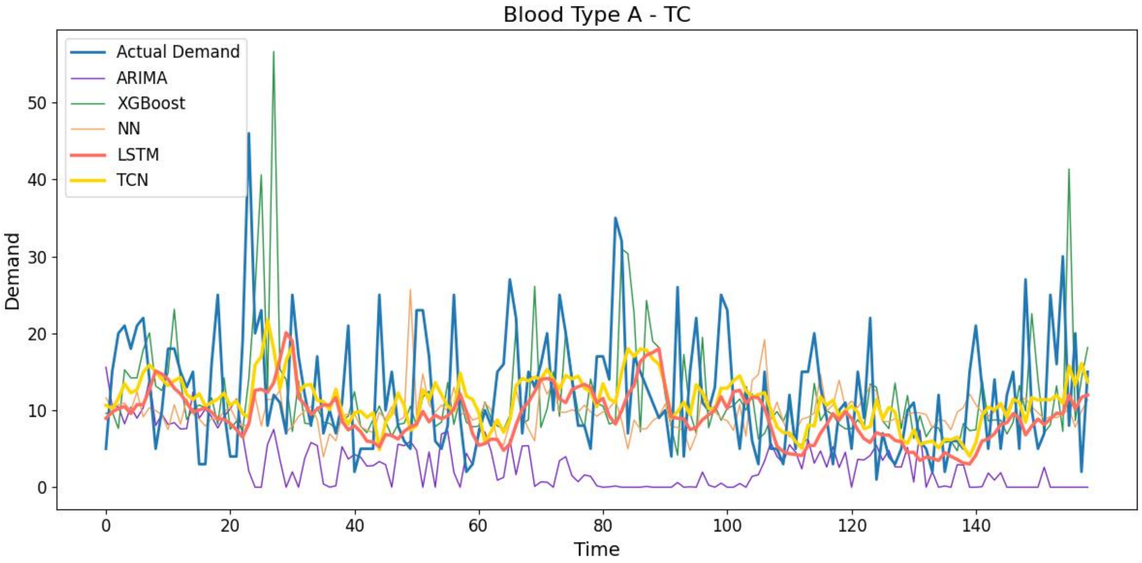


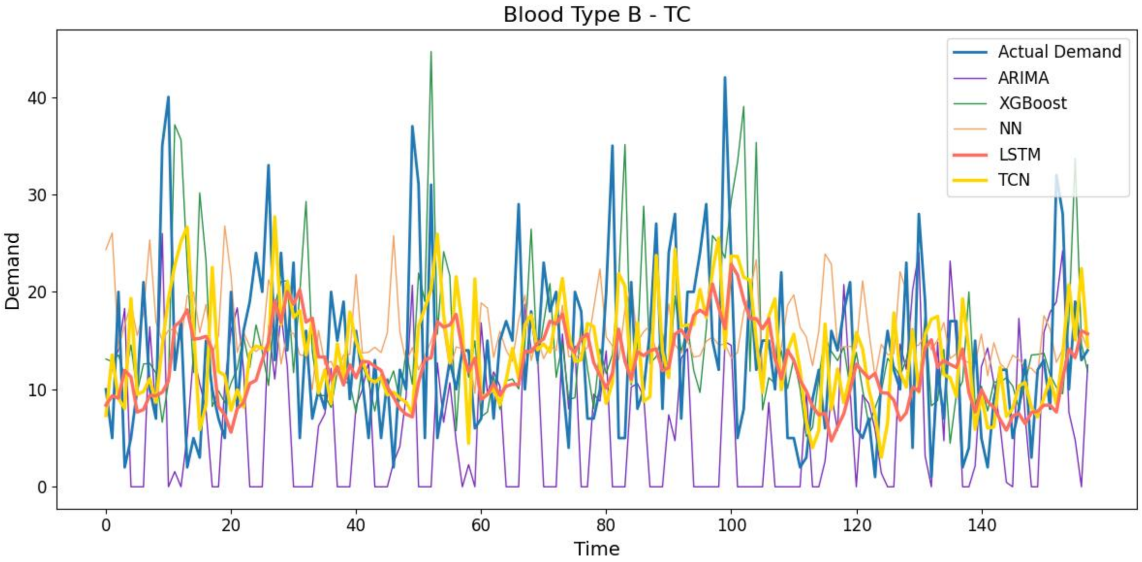

Blood Type AB - FFP
Actual Demand
ARIMA
XGBoost
NN
LSTM
TCN
Demand
0
5
10
15
20
25
30
Time
0
10
20
30
40

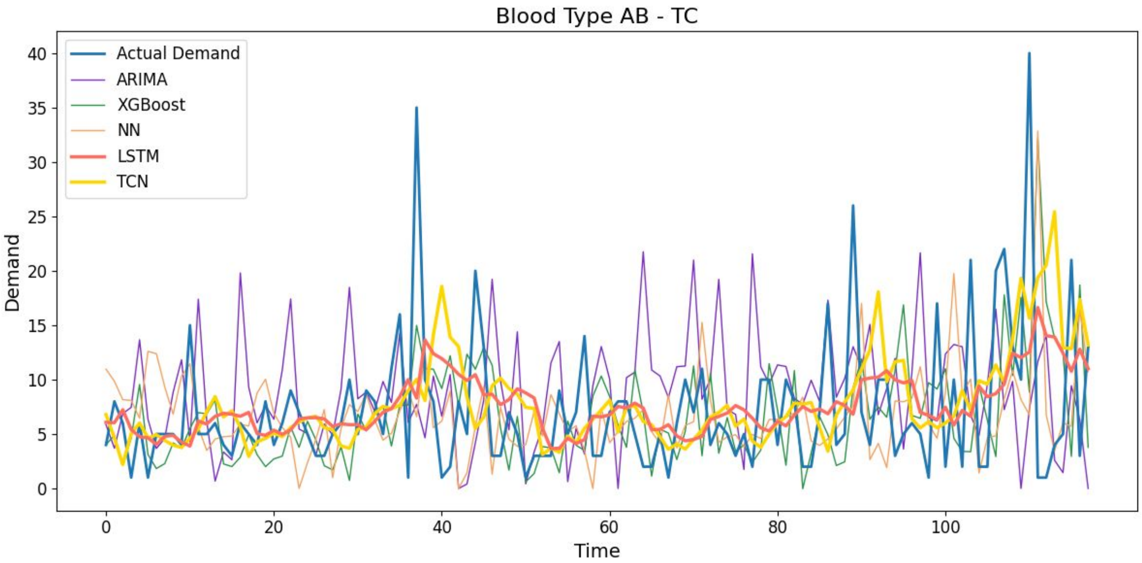

Blood Type A - RBC
Actual Demand
ARIMA
XGBoost
NN
LSTM
TCN
Demand
0
20
40
60
80
100
Time
0
25
50
75
100
125
150
175


Blood Type B - RBC
Actual Demand
ARIMA
XGBoost
NN
LSTM
TCN
Demand
0
20
40
60
80
100
Time
0
25
50
75
100
125
150
175


Blood Type AB - RBC
Actual Demand
ARIMA
XGBoost
NN
LSTM
TCN
Demand
0
10
20
30
40
Time
0
25
50
75
100
125
150
175


Blood Type A - TC
Actual Demand
ARIMA
XGBoost
NN
LSTM
TCN
Demand
0
10
20
30
40
50
Time
0
20
40
60
80
100
120
140


Blood Type B - TC
Actual Demand
ARIMA
XGBoost
NN
LSTM
TCN
Demand
0
10
20
30
40
Time
0
20
40
60
80
100
120
140



References

1. Basu D, Kulkarni R. Overview of blood components and their preparation. *Indian J Anaesth*. 2014;58(5):529–537. doi:10.4103/0019-5049.144647. PMID:25535413
2. Klein AA, Arnold P, Bingham RM, et al. AAGBI guidelines: the use of blood components and their alternatives 2016. *Anaesthesia*. 2016;71(7):829-842. doi: 10.1111/anae.13489. PMID:27062274
3. Sharma S, Sharma P, Tyler LN. Transfusion of blood and blood products: indications and complications. *Am Fam Physician*. 2011;83(6):719-724. PMID:21404983
4. Vibhute M, Kamath SK, Shetty A. Blood utilisation in elective general surgery cases: requirements, ordering and transfusion practices. *J Postgrad Med*. 2000;46(1):13-17. PMID:10855071
5. Belayneh T, Messele G, Abdissa Z, Tegene B. Blood Requisition and Utilization Practice in Surgical Patients at University of Gondar Hospital, Northwest Ethiopia. *Journal of Blood Transfusion*. 2013;2013:1-5. doi:10.1155/2013/758910. PMID:24369525
6. Arani M, Chan Y, Liu X, Momenitabar M. A lateral resupply blood supply chain network design under uncertainties. *Applied Mathematical Modelling*. 2021;93:165-187. doi:10.1016/j.apm.2020.12.010
7. Gilani Larimi N, Yaghoubi S. A robust mathematical model for platelet supply chain considering social announcements and blood extraction technologies. *Computers & Industrial Engineering*. 2019;137:106014. doi:10.1016/j.cie.2019.106014
8. Shih H, Rajendran S. Comparison of Time Series Methods and Machine Learning Algorithms for Forecasting Taiwan Blood Services Foundation’s Blood Supply. *Journal of Healthcare Engineering*. 2019;2019:1-6. doi: 10.1155/2019/6123745. PMID:31636879

9. Fanoodi B, Malmir B, Jahantigh FF. Reducing demand uncertainty in the platelet supply chain through artificial neural networks and ARIMA models. *Computers in Biology and Medicine*. 2019;113:103415. doi:10.1016/j.compbiomed.2019.103415. PMID:31536834
10. Moslemi AA, Attari MYN. Prediction of demand for blood bank products according to blood groups by a data mining approach by using neural networks. *International Journal of Hospital Research*. 2021;10(4).
11. Li N, Chiang F, Down DG, Heddle NM. A decision integration strategy for short-term demand forecasting and ordering for red blood cell components. *Operations Research for Health Care*. 2021;29:100290. doi:10.1016/j.orhc.2021.100290
12. Sun X, Xu Z, Feng Y, et al. RBC Inventory-Management System Based on XGBoost Model. *Indian J Hematol Blood Transfus*. 2021;37(1):126-133. doi:10.1007/s12288-020-01333-5. PMID:33707845
13. Ben Elmir W, Hemmak A, Senouci B. Smart Platform for Data Blood Bank Management: Forecasting Demand in Blood Supply Chain Using Machine Learning. *Information*. 2023;14(1):31. doi:10.3390/info14010031
14. Wang Y, Zhang W, Rao Q, et al. Forecasting demands of blood components based on prediction models. *Transfusion Clinique et Biologique*. 2024;31(3):141-148. doi:10.1016/j.tracli.2024.04.003. PMID:38670448
15. Motamedi M, Dawson J, Li N, Down DG, Heddle NM. Demand forecasting for platelet usage: From univariate time series to multivariable models. Jekarl DW, ed. *PLoS ONE*. 2024;19(4):e0297391. doi:10.1371/journal.pone.0297391. PMID:38652720
16. Miri-Moghaddam E, Bizhaem SK, Moezzifar Z, Salmani F. Long-term prediction of Iranian blood product supply using LSTM: a 5-year forecast. *BMC Med Inform Decis Mak*. 2024;24(1):213. doi:10.1186/s12911-024-02614-z. PMID:39075453
17. Wentz VH, Maciel JN, Gimenez Ledesma JJ, Ando Junior OH. Solar Irradiance Forecasting to Short-Term PV Power: Accuracy Comparison of ANN and LSTM Models. *Energies*. 2022;15(7):2457. doi:10.3390/en15072457
18. Sood S. A comparative performance of LSTM, ANN and ARIMA for prediction of stock price. In: *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*. IEEE; 2023:1-5. doi:10.1109/ICSES60034.2023.10465401
19. Pei Y. LSTM-TCN ensemble learning based photovoltaic power prediction. In: Siano P, Zhao W, eds. *Ninth International Symposium on Advances in Electrical, Electronics, and Computer Engineering (ISAEECE 2024)*. SPIE; 2024:284. doi:10.1117/12.3034420
20. Ghimire S, Deo RC, Casillas-Pérez D, Salcedo-Sanz S, Acharya R, Dinh T. Electricity demand uncertainty modeling with Temporal Convolution Neural Network models. *Renewable and Sustainable Energy Reviews*. 2025;209:115097. doi:10.1016/j.rser.2024.115097
21. Bernacki J. Forecasting the air pollution concentration with neural networks. *Urban Climate*. 2025;59:102262. doi:10.1016/j.uclim.2024.102262

22. Liu J. Navigating the Financial Landscape: The Power and Limitations of the ARIMA Model. *HSET*. 2024;88:747-752. doi:10.54097/9zf6kd91

23. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM; 2016:785-794. doi:10.1145/2939672.2939785

24. Mahadevan S, Poornima S, Tripathi K, Pushpalatha M. A survey on machine learning algorithms for the blood donation supply chain. *J Phys: Conf Ser*. 2019;1362(1):012124. doi:10.1088/1742-6596/1362/1/012124

25. Tu JV. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*. 1996;49(11):1225-1231. doi:10.1016/S0895-4356(96)00002-9. PMID:8892489

26. Arifin S, Wijaya A, Nariswari R, et al. Long Short-Term Memory (LSTM): Trends and Future Research Potential. *IJETAE*. 2023;13(5):24-34. doi:10.46338/ijetae0523_04

27. Bharilya V, Kumar N. Machine learning for autonomous vehicle's trajectory prediction: A comprehensive survey, challenges, and future research directions. *Vehicular Communications*. 2024;46:100733. doi:10.1016/j.vehcom.2024.100733

28. Schneider P, Xhafa F. *Anomaly Detection and Complex Event Processing over IoT Data Streams: With Application to eHealth and Patient Data Monitoring*. Academic Press, an imprint of Elsevier; 2022.

Sample Paper 3 - Blockchain for Supply Chain Management

Journal of Big Data

RESEARCH Open Access

Designing a Permissioned Blockchain Network for the Halal Industry using Hyperledger Fabric with multiple channels and the raft consensus mechanism

Isti Surjandari*, Harman Yusuf, Enrico Laoh and Rayi Maulida

*Correspondence:
isti@ie.ui.ac.id
Industrial Engineering Department, Faculty of Engineering, Universitas Indonesia, Kampus UI, Depok 16424, Indonesia

Abstract

Halal Supply Chain Management requires an assurance that the entire process of procurement, distribution, handling, and processing materials, spare parts, livestock, work-in-process, or finished inventory to be well documented and performed fit to the Halal and Toyyib. Blockchain technology is one alternative solution that can improve Halal Supply Chain as it can integrate technology for information exchange during the tracking and tracing process in operating and monitoring performance. This technology could improve trust, transparency, and information disclosure between supply chain participants since it could act as a distributed ledger and entitle all transactions to be completely open, yet confidential, immutable, and secured. This study uses a Blockchain Network with three channels and uses raft consensus algorithm in designing web interfaces and testing their capabilities. From the web interface, there were no failures in the validity test during the invoke test and the query test. In addition, the web interface was also successfully tested to thwart the formation of a block in case of data input errors from the user. The server can also do the process as a provider of information and validator for the web interface. From the results of simulations conducted on the Blockchain Network that was made, Blockchain's transaction speed is fast and all the transaction is successfully transferred to other peers. Thus, Permissioned Blockchain is useful for Halal Supply Chain not just because it can secure transactions from some of the halal issues, but the transaction speed and rate to transfer data are very effective.

Keywords: Blockchain, Hyperledger Fabric, Multiple channels, Raft, Halal Supply Chain

Introduction

The advent of the Fourth Industrial Revolution promises significant opportunities and challenges in many industries, one of them being in the supply chain. Supply chain industries are embracing automation and data exchange and implement new technologies including Blockchain, Artificial Intelligent, and Internet of Things (IoT) devices [1]. These innovations are fundamentally changing supply chain dynamics including Halal Industry.

Islamic economic development report published by the Dubai International Financial Center in 2019, states that the Halal Industry has increased rapidly compared to some other industrial sectors, increasing with an average growth of 100 billion US $ annually and is expected to reach US $ 3.2 Trillion in 2024 [2]. This growth is influenced by the increase of number of the Muslim populations in the world. From a report on Muslim growth published by the Pew Research Center in 2011, the estimated average growth of Muslims per 10 years is 1.63% and expected to be 26.4% of the total population the world in 2030 [3]. The size and growth of the Muslim population results in the increase of purchasing power, hence the value of the Halal Industry will increase.

Despite Halal Industry's increase in purchasing power and value, the condition has not yet achieved its optimal potential as there are still many sectors that can be improved. For example, although Indonesia is recognized as Muslim's world's most populous country, it lags behind other Muslim-majority countries in creating an ecosystem that supports Halal Industry according to the Global Islamic Economy Indicators (GIEI) 2019. Thus, there are still many potentials that can be improved to maximize the Halal Industry both domestically and globally, such as improving Halal Supply Chain quality [4].

The government roles in improving Halal Supply Chain is substantial to enforce a law that requires all business actors to make Halal Certificates on food products, medicines, cosmetics, and other genetically engineered products. Halal Certificate is a proof or guarantee that the products are safe and acceptable in accordance with Islamic law. Yet, Halal Certificate is inadequate to improve the Halal Supply Chain. Customers must ensure that the raw material used and the process to make a product is halal, as well as the final product. Other way to improve the Halal Supply Chain is to integrate technology for information exchange during the tracking and tracing process in operating and monitoring performance [5]. Moreover, both vertical and horizontal collaborative relationships in the form of trust, transparency, and information disclosure between supply chain participants is essential to maximize the integration of technology and information with the Halal Supply Chain and increase mutual effectiveness and efficiency [6].

This paper aims to adopt a blockchain framework for Halal Supply Chain case by using Hyperledger Fabric. In addition, this paper will test it to find out its capability to Halal Supply Chain in found out some blockchain key aspect that can improve Halal Supply Chain, validating the transaction process, and testing the transaction speed. The rest of this paper is organized as follows: The "Literature review" section consists of some theoretical concept. The "Methodology" section consists of some related work of the method that will be used. The "Result" section consists of the adopted Blockchain Framework and its overview of the finished interface of blockchain framework. The "Discussion" section consists of the discussion of the result obtained in this research. The "Conclusion" section consists of this research conclusion and some possible suggestions for this research.

However, the empirical results reported herein should be considered in the light of some limitations. First, the data used to design the Blockchain architecture is arranged based on the Halal Supply Chain flow in the study by Simatupang et al. [7]. Therefore, some adjustments are needed if the model adopted for a different flow. Furthermore, Hyperledger Fabric version 1.4.3 is used to design the blockchain architecture. Thus, the advancement of the software in the future could result better model performance than the current model.

Literature review

In this part, the theoretical basis and the concepts used in this study are discussed. The concepts explained include theories and applications about Halal Supply Chain, Blockchain, Permissioned Blockchain, and Hyperledger Fabric.

Halal Supply Chain Management

Halal Supply Chain Management may be defined as a network in assuring the entire process of procurement, distribution, handling, and processing materials, spare parts, livestock, work-in-process or finished inventory to be well documented and performed fit to the Halal and Toyyib [6]. Halal itself is something that is permitted in accordance with the rules that already exist in the Qur'an and the Hadith, while the term Toyyib means healthy and good [8]. The Toyyib concept can also be used to enrich society with spiritual, moral, and humanitarian values, as well as food safety regulations [9]. However, the gray area (located between halal and haram) causes doubt in the application of the Halal concept. Therefore, the opinion of the academic, religious regulations (fatwas), and local customs in assessing and determining the product is needed to determine which product is prohibited or allowed to be consumed [10].

Halal Supply Chain carry five fundamental issues, which are traceability (ability to discover information about location and origin of the product); regulation for product withdrawal related to halal prerequisites; end-to-end Halal Supply Chain integrity from producer into customer; contradictory systems and different interpretations regarding Halal Supply Chain; and lack of integration of technology and information with the Halal Supply Chain [11].

Blockchain

Blockchain technology is one of the alternative solutions that can improve Halal Supply Chain. This technology could resolve these problems since it could act as a distributed ledger and entitle all transactions to be completely open, yet confidential, immutable, and secured. Blockchain provides security as protection and prevention from duplication, or distortion of data from outside noise. The participants (in decentralized-computer-terminal form) are connected by using key-access system enabling direct transactions between sellers and buyers without intermediaries [12]. Because blockchain nature is a distributed ledger database, there are many things that can be improved by using blockchain such as big data for data analysis [13, 14].

Supply Chain transactions will be gathered in a set of blocks when each set of new transactions is added successfully. The block will be added to the Blockchain Network in a linear chronological order with timestamp [12]. Each supply chain participant, known as peer node on Supply Chain Network, receives a copy of blockchain which can be downloaded automatically. Peer nodes would have access to all information which includes supply chain participant's address and supply chain path, hence even the user will know the flow of the manufacturing process of a particular product [15].

Permissioned Blockchain

Permissioned Blockchain, which is a specialized form of blockchain for private transactions and having sufficient speed of transactions in real time basis [16, 17], is the most suitable type of Blockchain Network for Halal Supply Chain. The role of each Supply Chain participant will be determined by the administrator to decide what information can be seen and added. Intrinsic configuration of the blockchain manages transaction nodes and defines the role of the nodes in accessing or making changes to the Blockchain, including maintaining the identity of each Supply Chain participant in the Blockchain Network [18]. Supply chain participants such as suppliers, distributors, wholesalers, and retailers focus solely on their respective parts. Thus, regulators are essential to determine the role of each supply chain participant even though in determining each role, must also be made by consensus so that no one feels disadvantaged [16, 19].

Halal Supply Chain and Blockchain

Halal Supply Chain needs transparency thus authenticity and reliance of halal brands can be ensured. Blockchain combines distributed ledgers and smart contracts so that the performance of the Halal Supply Chain will be increased. The improvement will generate more dependable information and assurance of Halal Supply Chain; smooth and effective halal process from beginning of production process to consumer's point of purchase; Halal Supply Chain sustainability; consumer trust in the halal brand; and acknowledgment from worldwide of the halal Blockchain [6].

Fundamental principle of Halal Blockchain is to combine all different Mazhab in targeted markets with Islamic schools, religious regulations (fatwas), and local traditions. Halal Blockchain must be pertinent for all countries (be it Muslim or non-Muslim). Halal Supply Chain participants are given information automatically about the process compliance based on specific product market scenarios. Halal Blockchain's authenticity and security is a priority to secure confidential data and minimize the opportunities of cyber-attack [5].

Halal Blockchain gives some benefit to producers, distributors, retailers, logistic service providers, and halal certification agencies. Halal certification agencies must adopt Blockchain technology to gain more trust and authenticity of the halal brand. They need to support halal certification of all Halal Supply Chain Instances to encourage the application of more obedient transportation and warehousing downstream the Supply Chain. Harmonizing the standards of Halal Supply Chain in various countries will be critical to support the Halal Industry and their global supply chain [5, 11].

Prior research related to Blockchain and Halal Supply Chain stated that there are three issues faced by the Halal Supply Chain globally, which are: contamination, disobedience, and perception. In this case, Blockchain technology is potential in resolving the first two problems (i.e., contamination and disobedience). However, the application of Blockchain needs to combine with Halal Certificate from each Supply Chain participant to get a better outcome [6].

Hyperledger Fabric

Hyperledger Fabric is an open-source Distributed Ledger Technology platform which is widely used for various company-related cases. This platform is very interactive to create a blockchain framework due to its modular and configurable architecture. The explanation of Hyperledger Fabric documentation is also very comprehensive compared to other platform and there are many developers contributed to developing this platform.

Hyperledger Fabric V.1.x distinguish the transaction into two types, execution transaction and ordering transaction. Whereas, there are three steps of transaction flow, which are execution, order, and validation. Each transaction can be executed in separated peer and can be executed before consensus from the ordering service is executed [20].

In the Blockchain system, there are some key terms such as nodes, data structures, transactions, ordering services, and channels [16]. Blockchain Networks must consist of several nodes. These nodes are usually defined as a virtual entity because it could run on physical hardware. Peers, orderers, and clients, in general are a set of nodes in the Blockchain Network [21]. Peers make transactions and distribute ledgers. In general, all peers are committers. On the other hand, orderers keep all the orders from the transaction that has been committed, creating new blocks, and search for consensus.

Clients are a set of nodes that act as end users of Blockchain Networks. The roles of client are sending a transaction proposal to peers, coordinating the results of the execution, verifying whether the transaction is valid, and sending the transaction that has been verified by peers to the ordering service. Furthermore, the data structure maintains global status in all associates using key value storage and ledgers (KVS). KVS manages and maintains the system to be updated, while the ledger provides a valid and verified history of all state changes [21].

In Hyperledger v1.4.x there are several types of transactions, such as init (deploy), invoke, and query. Init or deploy is useful for installing and instantiating chaincodes hence the transactions can be run. While invoke are useful for invoking transactions from chaincodes that have been installed and instantiated, query can be used for checking what transactions were successfully carried out in the process [22].

The transaction flow on v1 fabric follow the following steps:

1. The client makes a transaction and sends it to all endorser peers according to the chain.
2. Each endorser peers authorizes transaction execution and makes endorsement signatures.
3. Clients collect support signatures from endorser peers and collect them through the ordering service.
4. Ordering services create transaction blocks and maintain orders with a timestamp.
5. When supporting partners receive a block of transactions, they will assess the transaction against its authorization policy, then determine the validity of the transaction.

Ordering Services Nodes provided by Hyperledger Fabric have the role of managing and maintaining channel configuration as well as executing the transaction process. In the channel configuration section, ordering services nodes has the power to control the basic channel access section along with a consortium, which is configured in advance through the configuration file. Ordering service nodes can control which nodes, according to the previously defined consortium, are able to read and write transaction data [23].

Ordering Services Nodes also have some functions for each process of transaction flow (Order, Execute, Validate). In the Order Phase, the ordering service node will collect endorsed transactions (transactions that have already been endorsed by endorsing peers) from clients. After the collected transaction process hit batchSize (the limit of transactions that can be collected per batch) or batchTimeout (the time limit for collecting transactions per batch) [24], ordering service nodes will set batch transactions in a strict order and turn them into a block in the execute phase. Since transactions inside the block are in strict order, all successful and validated transactions will not be thrown away (there will be no ledger forks). Last but not least, in the validation step, the order will allocate blocks to all peers that are connected to the same channel (depending on the configuration of the channel) [23].

There are three types of Ordering Services Nodes Implementation, which are Solo, Kafka, and Raft:

1. Solo

Solo is one of Ordering Service Implementation to evaluate the Blockchain that has been developed. Solo operates without a consensus algorithm and contains only one ordering node [23].

2. Kafka

Kafka is one of the ordering service implementations originated from the Crash Fault Tolerant (CFT), where the process can proceed even though some of the current nodes encounter N failures while $N/2+1$ nodes still able to run [24]. This consensus mechanism uses "leader and follower" in the configuration node and is handled by Zookeeper Ensembled. However, the method of seeking offset numbers is from the ordering service node (the ordering service node has already been configured to preserve local logs) and not via Kafka partitions like the usual process of kafka. The process is slower than straight from Kafka but duplication of the block is unlikely to occur [10, 22, 23].

3. Raft

In fact, Raft is similar to Kafka because the implementation of the ordering service also uses CFT. Raft uses the Raft consenter as ordering services nodes to implement the "leader and follower" process. The raft is also used by Hyperledger Fabric as a bridge connector to create a consensus of Practical Byzantine Fault Tolerant since they have a similar procedure in the integration of Hyperledger Fabric [23, 25].

Kafka and raft have the same consensual mechanism, but there is an apparent difference in the operation of the two ordering service node implementations. These aspects are the main reason for driving the use of Raft rather than Kafka in this paper.

1. General comparison

Kafka and Zookeeper not compatible for massive networks. However, there are several organizations and channels on the Blockchain Network, the mechanism almost like one organization only which is not too decentralized. Raft on the other hand, uses ordering service nodes as a state replication machine (Raft Consenter) directly. Thus, all organizations in the Blockchain Network will have their own ordering service node and the Blockchain Network will be more decentralized [23].

Kafka also requires docker images to run for the CFT since Kafka was developed by Apache. This is overly complex, and its application needs to be further studied. On the other hand, Raft was natively developed by Hyperledger Fabric itself to make it easier [23].

2. Difference in terms of configuration
   Overall, Raft is simpler than Kafka in terms of configuration. Raft is designed directly from the ordering service node [25] while Kafka must use Kafka brokers and Zookeeper Ensemble to make CFT process work [22]. However, Raft is more difficult than Kafka when configuring the individual channel because Raft must set up the transport layer security (TLS) certificate for client and server [25]. While Kafka only needs to decide the number of Kafka brokers and Zookeepers [22].
   When building a docker container, Raft only uses ordering services nodes that have already been configured by the previous network configuration [25, 26]. On the other hand, Kafka must separate the work of Kafka and Zookeeper containers and must specify the amount of Kafka and Zookeeper containers in the docker compose file. That is why the process of running CFT using Kafka is getting trickier [22].
   All nodes, such as peers, can interact with other peers by using channels or using private data. The channel is private in terms of making transactions; only users who are on the same channel can only make transactions. Yet, users in a different channel can see the data due to the transparency concept. In contrast, private data makes transactions private in the channel and specific peers. Only peers that are already configured with it can do the transactions, even though it will eventually be distributed to others when the block is distributed [27].

Methodology

This part consists of some related work of the methodology that will be used with a case study example.

Research object

For the purpose of this study, a halal meat industry example based on some research will be used. There are several reasons for choosing a halal meat industry as a sample. In the halal meat industry, to keep the "Toyyib" exists from its own characteristics is wildering due to its perishable characteristic and difficulty to control the temperature [28]. These characteristics become a challenge for the supplier, producer, distributor, wholesaler, and retailer to keep the safety and freshness of fresh meat from contamination, disobedience, and perception. Blockchain can handle the first two problems, contamination, and disobedience because of its aspects as explained in the previous part [10].

The supply chain participants of halal meat industry include suppliers, producers, distributors, and customers. In halal meat industry, only livestock that meet certain criteria of age will be slaughtered. The meat will be sent to the producers to be processed and cooled. Then, it will be tested to determine whether it is halal or otherwise. Only the halal-certified meat will be sent to the distributor. The distributors will send it to the end customer or else be refrigerated [7].

In general, the supply chain participants can easily see the halal certification from the regulator. However, since customers only see the final product, not the raw material used nor the process to make a product to see if it is halal, there are some of them who are doubting the halalness of Halal Industry chain. A diagram of the halal supply chain flow can be seen in Fig. 1.

The Blockchain Network of meat ordering system was created to resolve the traceability problems of the final product. Blockchain as a distributed ledger is obliged to record the transactions in a time sequence. After the transaction is gathered in a set of blocks, the block is irrevocable. The irrevocable block characteristic is feasible to resolve customer's doubt on the traceability of halal-certified meat.

In this study, the Blockchain Network consists of three channels, with each channel consisting of two organizations, and each organization has one client and two peers. The first channel is to connect suppliers and producers, the second channel is to connect producers and distributors, and the third one to connect distributors and end users. Three channels are designed separately because each participant may have a different price agreement thus their privacy is protected. The Blockchain Network also consists of five orderer nodes that have been configured with the Raft consensus mechanism. The channel configuration diagram can be seen in Fig. 2.

For simplicity and due to similarity in the flow process of each channel, this research will explain the configuration process of the first channel, which is the channel for supplier and producer. The steps for configuring the Blockchain Network consists of [29]:

1. Creating the Blockchain Network for all Halal Supply Chain entities and network administrators (which is needed once in the process because all channels must be included in the same network to make some peers have the ability to interact in multiple channels).
2. Defining the consortium of the entities (for example, the supplier will be bound with the producer in the consortium because they will do the transaction process).

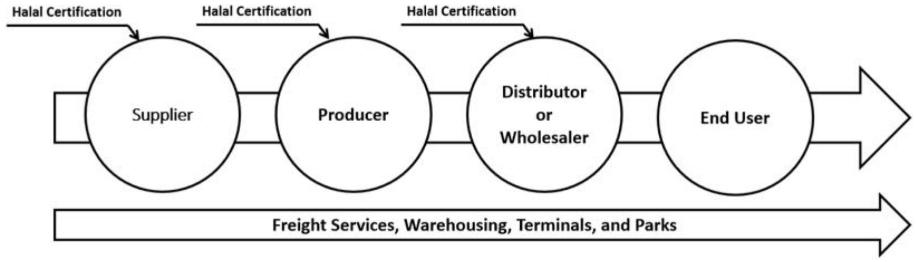


Fig. 1 Ordering Service Process on Halal Supplier to End-User [7]

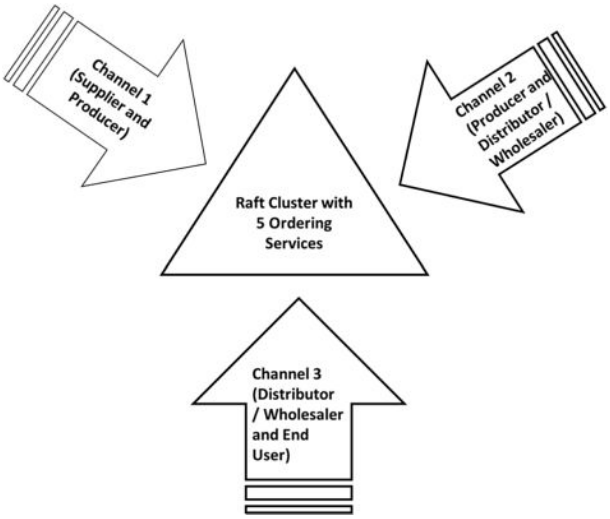

Channel 1 (Supplier and Producer)
Channel 2 (Producer and Distributor / Wholesaler)
Raft Cluster with 5 Ordering Services
Channel 3 (Distributor / Wholesaler and End User)

Fig. 2 Blockchain framework for Halal Supply Chain

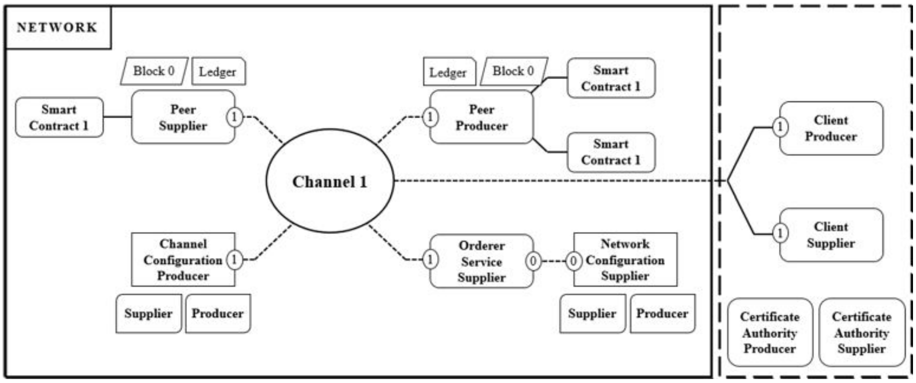


Fig. 3 The Result of Blockchain Channel for Supplier and Producer in The First Channel of Blockchain Network [29]

3. Creating the first channel for the consortium process of supplier and producer.
4. Inserting peers of supplier and producer into the channel (to interact with other peers and save the transaction proof on the ledger).
5. Installing and instantiating the Chaincode to each peer (the supplier just needed once in the process but the producer twice because the producer will be added to the other channels to which is the channel for producer and distributor). The reason is although the smart contract in this design is similar, the real-world process actually different because the contract in the real world should be more complicated.

The result of the blockchain configuration process can be seen in Fig. 3.

Result

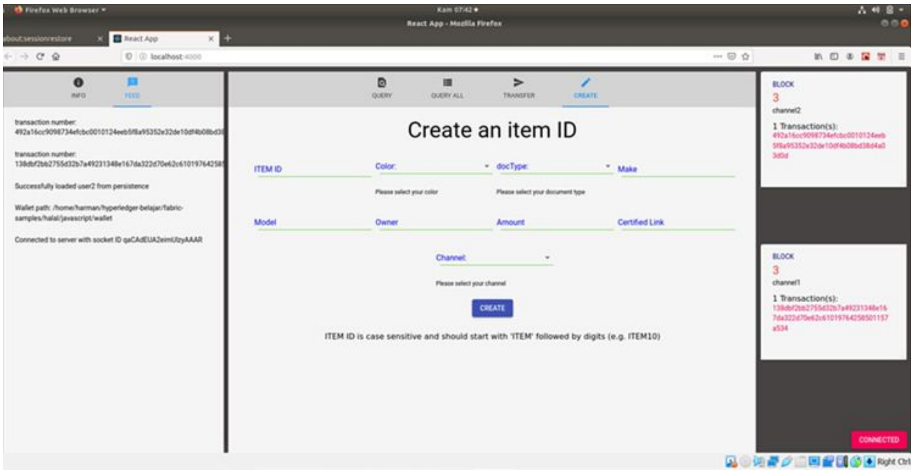
In this chapter, the overview of the Blockchain Network server and web interface will be discussed.

Overview of the Blockchain Network server and Interface

This blockchain is basically a development of fabcar with its chaincode is modified according to the Halal Supply Chain. There are four processes of the Blockchain prototype used on the Halal Supply Chain. The process involves querying a single item ID (checking one item in the Blockchain), querying all items (check all items in the Blockchain), transfer an item ID (transfer one or more items to another company), and create an item ID (create one or more items in the Blockchain). The main change from the chaincode is the Create an Item ID process. The shape of the web interface can be seen in Fig. 4.

The invoking process defines as a process of entering transaction data into a Blockchain. This process includes Create an Item ID and Transfer an Item ID. Create an item ID has several special attributes consisting of item ID, color, doctype (the type of item to be sent), make (item name), model (specification/type of item), the owner (owner), amount (amount goods to be transferred), and certified links (proof of halal certification links). There are special attributes owned by some companies for making transactions on different channels, which is the channels attribute. There are also some of the attributes for creating an item ID that is being changed or added from fabcar chaincode, such as are Color, docType, Amount, Certified Link, and Channel.

In the color attribute, the user can choose what color to add to the attribute. This setting will be saved hence the user does not need to write the type of the item color again. It is more convenient for the user if they want to input the historical data so that they do not need to type it from the beginning again. In the docType attribute, created or transferred items in the Blockchain are not only in the form of food, but drinks or items in general can also be included in the Blockchain. Thus, on the Amount attribute, the company that wants to create an item can tell how many items will be sent. This prevents repetition on entering data if it turns out that the items sent are in large quantities. For created or transferred items that are considered to be halal, the company needs to attach the halal certification link. Trust between each supply chain participant about the items in halal category is expected to increase with this attribute.



transaction number:
492a16cc9098734efcbc0010124eeb5f8a95352e32de10df4b08bd3

transaction number:
138dbf2bb2755d32b7a49231348e167da322d70e62c610197642581

Successfully loaded user2 from persistence

Wallet path: /home/harman/hyperledger-belajar/fabric-samples/halal/javascript/wallet

Connected to server with socket ID qaCAdEUA2eimUIzyAAAR

Create an item ID

ITEM ID

Color:
Please select your color

docType:
Please select your document type

Make

Model

Owner

Amount

Certified Link

Channel:
Please select your channel

CREATE

ITEM ID is case sensitive and should start with 'ITEM' followed by digits (e.g. ITEM10)

BLOCK
3
channel2
1 Transaction(s):
492a16cc9098734efcbc0010124eeb5f8a95352e32de10df4b08bd38d4a03d0d

BLOCK
3
channel1
1 Transaction(s):
138dbf2bb2755d32b7a49231348e167da322d70e62c61019764258501157a534

Fig. 4 Display of Blockchain Network's Web Interface

Furthermore, all companies that enter two channels (such as producers who interact with suppliers and distributors or distributors who interact with producers or customers) have special attributes in the interface, called channels. There are two advantages of this channel attribute which are: (1) items that the company wants to create or transfer to another company will be sent correctly and (2) The interface of a company that has two channels is not necessarily separated into two interface so that it is easier and more interactive in conducting transactions through the company.

To prove that the system can be made, there are two key that must be considered, which are the web interface [can block data be created directly from the web interface (client)] and the integrity between the web interface and server of the terminal (whether the data is truly properly stored on the server so that data is not lost).

Discussion

In this chapter, we will discuss the validity test of the web interface, and test the integrity of the web interface and server to test whether the web interface of the blockchain can be used as well as a test of the Blockchain Network transaction simulation to find out how many transactions can be received by the Blockchain Network.

Validity of the web Interface

When the web interface is successfully initiated into the web browser, the block and transaction ID data of the block will appear on the right. It indicates that the transaction data is successfully stored in the Blockchain. In the previous image we can see that the third block of the Blockchain was formed before the web interface was used. This is because of the process of installing and instantiating the chaincode that has been done previously to initiate the web interface. When all data attributes are successfully entered according to their respective attributes, the latest block (in this case, block 4) and the transaction id will automatically appear on the right side of the web interface. Moreover, the left side of the web interface will notify the user that the input of data item from the user's company is successfully done.

The process of making a block can only be done on the create an item ID and transfer an item ID. This is due to the process of writing and reading data (the process of writing data so that new data appears or changing old data) occurs in both parts. The process of writing new item data can be done to create an item ID and the process of changing item ownership is in the transfer of an item ID. Both processes must be separated even though they have the same goal of entering data into the Blockchain. Creating an item ID can only be done if the data item entered in the data form has an id form that is still not registered in the company's Blockchain database. Meanwhile, the transfer of an item ID can only be done if the item data changed ownership is already registered in the Blockchain database. This is because the main purpose of the transfer of an item is to change the state of the database as well (the most recently viewed data), thus creating a new block. Old blocks of data item ID whose ownership has not been changed will remain the same to maintain the Blockchain's tampered-proof nature.

Meanwhile, from querying an item ID and querying all item ID, the two parts of the web interface have a function as reading data (only to check the items in the Blockchain and their data attributes). When both parts of the web interface are executed, the new block on the right will not be formed because its main purpose is only to check the block and transaction ID. But on the left side of the web interface, state database information (the most updated information) about the item and its attributes will appear on demand. The difference between the two parts is that the query an item ID will only search for one specific item ID that matches the channel that connects the information while the query all item ID looks for all information about the item ID corresponding to the channel that connects the information.

There are several cases that cause a block to be filled with incorrect data and cause the Blockchain to be invalid, such as incomplete entering data attributes and incorrect data entry. In order to prevent the problem, interfaces and servers are created using prevention tools such as pop ups or errors so that invalid data will not be formed in a company's Blockchain database server. For example, in creating an item ID section, the first requirement is that all data must be entered. If there is at least one part of the data attributes that is not entered, then the item data cannot be entered into the Blockchain channel.

In addition, in the Item ID section, filling in the data form can only be done if the first four letters are "ITEM" in capital letters and followed by a number (for example: "ITEM4"). Then, the item ID must be different from the item ID that exists on the Blockchain so that there is no duplication in the Blockchain item data. If all the requirements above are not met, an error will appear on the interface. Unlike creating an item ID, transferring an item only allows existing Item ID data. If the item ID on an item transfer is apparently not in the Blockchain, an error will appear reminding the user that the data is still not in the Blockchain so the item transfer process cannot be performed.

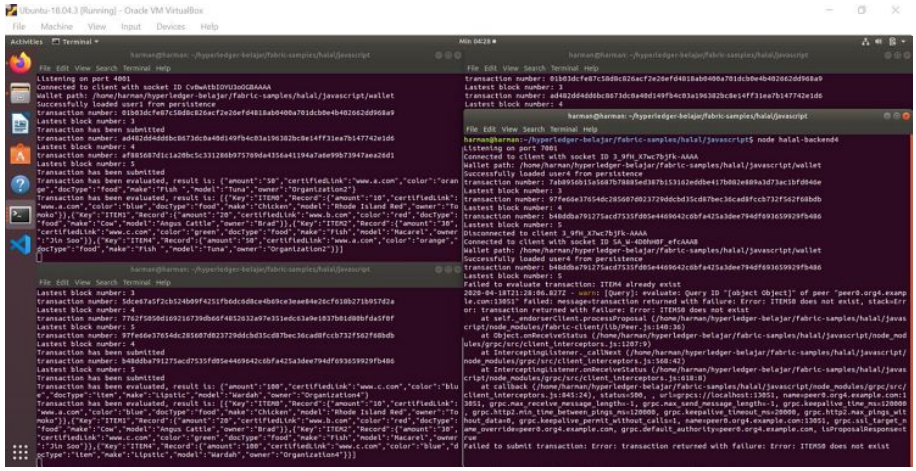
Integrity of web Interface and server

There are other functions regarding the server besides storing transaction data from the web interface. The terminal server also functions as a notification for the company even though the web interface is still not turned on. When the company wants to check information about any item ID already in the Blockchain through the web interface, that information will also appear on the terminal server. Then, when the company sends data or change data about the item ID into the Blockchain through the web interface, the terminal server will notify the company that sent the notification that the information sent was successfully saved to the Blockchain and send information about the number of blocks that have been created in the Blockchain. The terminal server also sends notifications to companies that are on the same channel as companies that send data about item ID or change data about item ID that data changes occur within the Blockchain and send information about the number of blocks that have been created in the Blockchain. Finally, if there is an error in inputting data, the terminal server will notify that there is an error that one of the data attributes sent is not in the format. The process can be seen in Fig. 5.

Transaction simulation test

Simulation configuration

The blockchain is tested for four rounds and five replications with a total of 60 rounds. Each channel is tested for two rounds for the invoking process (create an item ID and


Ubuntu-18.04.3 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Min 04:28
harman@harman: ~/hyperledger-belajar/fabric-samples/halal/javascript
File Edit View Search Terminal Help

```
Listening on port 4001
Connected to client with socket ID CvBwAtbIOYU3oOGBAAAA
Wallet path: /home/harman/hyperledger-belajar/fabric-samples/halal/javascript/wallet
Successfully loaded user1 from persistence
transaction number: 01b03dcfe87c58d8c826acf2e26efd4818ab0400a701dcb0e4b402662dd968a9
Lastest block number: 3
Transaction has been submitted
transaction number: ad482dd4dd6bc8673dc0a40d149fb4c03a196382bc8e14ff31ea7b147742e1d6
Lastest block number: 4
transaction number: af885687d1c1a20bc5c331286b975769da4156a41194a7a6e99b73947aea26d1
Lastest block number: 5
Transaction has been submitted
Transaction has been evaluated, result is: {"amount":"50","certifiedLink":"www.a.com","color":"oran
ge","docType":"food","make":"Fish ","model":"Tuna","owner":"Organization2"}
Transaction has been evaluated, result is: [{"Key":"ITEM0","Record":{"amount":"10","certifiedLink":
"www.a.com","color":"blue","docType":"food","make":"Chicken","model":"Rhode Island Red","owner":"To
moko"}},{"Key":"ITEM1","Record":{"amount":"20","certifiedLink":"www.b.com","color":"red","docType":
"food","make":"Cow","model":"Angus Cattle","owner":"Brad"}},{"Key":"ITEM2","Record":{"amount":"30",
"certifiedLink":"www.c.com","color":"green","docType":"food","make":"Fish","model":"Macarel","owner
":"Jin Soo"}},{"Key":"ITEM4","Record":{"amount":"50","certifiedLink":"www.a.com","color":"orange","
docType":"food","make":"Fish ","model":"Tuna","owner":"Organization2"}}]
```

```
Lastest block number: 3
transaction number: 5dce67a5f2cb524b09f4251fb6dc6d8ce4b69ce3eae84e26cf610b271b957d2a
Lastest block number: 4
transaction number: 7762f5050d169216739db66f4852632a97e351edc63a9e1037b01d00bfda5f0f
Lastest block number: 5
transaction number: 97fe66e37654dc285607d023729ddcbd35cd87bec36cad8fccb732f562f68bdb
Lastest block number: 4
Transaction has been submitted
transaction number: b48ddba791275acd7535fd05e4469642c6bfa425a3dee794df693659929fb486
Lastest block number: 5
Transaction has been submitted
Transaction has been evaluated, result is: {"amount":"100","certifiedLink":"www.c.com","color":"blu
e","docType":"item","make":"Lipstic","model":"Wardah","owner":"Organization4"}
Transaction has been evaluated, result is: [{"Key":"ITEM0","Record":{"amount":"10","certifiedLink":
"www.a.com","color":"blue","docType":"food","make":"Chicken","model":"Rhode Island Red","owner":"To
moko"}},{"Key":"ITEM1","Record":{"amount":"20","certifiedLink":"www.b.com","color":"red","docType":
"food","make":"Cow","model":"Angus Cattle","owner":"Brad"}},{"Key":"ITEM2","Record":{"amount":"30",
"certifiedLink":"www.c.com","color":"green","docType":"food","make":"Fish","model":"Macarel","owner
":"Jin Soo"}},{"Key":"ITEM4","Record":{"amount":"100","certifiedLink":"www.c.com","color":"blue","d
ocType":"item","make":"Lipstic","model":"Wardah","owner":"Organization4"}}]
```

```
transaction number: 01b03dcfe87c58d8c826acf2e26efd4818ab0400a701dcb0e4b402662dd968a9
Lastest block number: 3
transaction number: ad482dd4dd6bc8673dc0a40d149fb4c03a196382bc8e14ff31ea7b147742e1d6
Lastest block number: 4
```

```
harman@harman:~/hyperledger-belajar/fabric-samples/halal/javascript$ node halal-backend4
Listening on port 7001
Connected to client with socket ID 3_9fH_X7wc7bjFk-AAAA
Wallet path: /home/harman/hyperledger-belajar/fabric-samples/halal/javascript/wallet
Successfully loaded user4 from persistence
transaction number: 7ab8956b15a5687b78885ed387b153162eddbe417b082e889a3d73ac1bfd04de
Lastest block number: 3
transaction number: 97fe66e37654dc285607d023729ddcbd35cd87bec36cad8fccb732f562f68bdb
Lastest block number: 4
transaction number: b48ddba791275acd7535fd05e4469642c6bfa425a3dee794df693659929fb486
Lastest block number: 5
Disconnected to client 3_9fH_X7wc7bjFk-AAAA
Connected to client with socket ID SA_W-4D0hH0F_efcAAAB
Wallet path: /home/harman/hyperledger-belajar/fabric-samples/halal/javascript/wallet
Successfully loaded user4 from persistence
transaction number: b48ddba791275acd7535fd05e4469642c6bfa425a3dee794df693659929fb486
Lastest block number: 5
Failed to evaluate transaction: ITEM4 already exist
2020-04-18T21:28:06.827Z - warn: [Query]: evaluate: Query ID "[object Object]" of peer "peer0.org4.examp
le.com:13051" failed: message=transaction returned with failure: Error: ITEM50 does not exist, stack=Err
or: transaction returned with failure: Error: ITEM50 does not exist
    at self._endorserClient.processProposal (/home/harman/hyperledger-belajar/fabric-samples/halal/javas
cript/node_modules/fabric-client/lib/Peer.js:140:36)
    at Object.onReceiveStatus (/home/harman/hyperledger-belajar/fabric-samples/halal/javascript/node_mod
ules/grpc/src/client_interceptors.js:1207:9)
    at InterceptingListener._callNext (/home/harman/hyperledger-belajar/fabric-samples/halal/javascript/
node_modules/grpc/src/client_interceptors.js:568:42)
    at InterceptingListener.onReceiveStatus (/home/harman/hyperledger-belajar/fabric-samples/halal/javas
cript/node_modules/grpc/src/client_interceptors.js:618:8)
    at callback (/home/harman/hyperledger-belajar/fabric-samples/halal/javascript/node_modules/grpc/src/
client_interceptors.js:845:24), status=500, , url=grpcs://localhost:13051, name=peer0.org4.example.com:1
3051, grpc.max_receive_message_length=-1, grpc.max_send_message_length=-1, grpc.keepalive_time_ms=120000
, grpc.http2.min_time_between_pings_ms=120000, grpc.keepalive_timeout_ms=20000, grpc.http2.max_pings_wit
hout_data=0, grpc.keepalive_permit_without_calls=1, name=peer0.org4.example.com:13051, grpc.ssl_target_n
ame_override=peer0.org4.example.com, grpc.default_authority=peer0.org4.example.com, isProposalResponse=t
rue
Failed to submit transaction: Error: transaction returned with failure: Error: ITEM50 does not exist
```

Fig. 5 Back end of Blockchain Network's web Interface

transfer an item ID) and two rounds for the query process (query an item ID and query all items). Because the simulation process is intended to test the maximum ability of Blockchain Network in receiving transactions, each test is tested using "fixed-backlog" to determine the maximum speed of TPS (transactions per second) and the maximum transactions that experience a backlog of 5 transactions. Each replication has a different transaction duration, which is 10, 20, 30, 40, and 50 s. The reason for choosing a simulation based on transaction duration rather than the number of transactions is the same as the reason why using the "fixed-backlog" type.

The Blockchain Network capability was made for previous research cases using three channels and crash tolerant crashes Raft crash tested with Ubuntu 18.04 LTS 64-bit 18 GB RAM 250 GB Hard Disk installed with hyperledger caliper and hyperledger fabric.

Simulation result

Based on Fig. 6, each channel in a blockchain model shows the same pattern regardless of the value of transaction speed. The send rate is relatively constant in each channel on each value speed of transaction, ranging from 32.0 to 49.0 TPS. However, the throughput value display two different patterns depending on the type of the process. Compared to another model, this model gives better outcome regards to the value of throughput rate with the relatively same hardware used for processing. Blockchain model proposed by Geneiatakis et al. resulted in throughput value ranging from 2.5 to 13 TPS [30], while the outcome of the blockchain model by Yusuf et al.. resulted in throughput value with an average of 27.9 TPS, the highest of 34.1 TPS and the lowest of 25.3 TPS [17].

The query process consists of query an item ID and query all items, resulting in a throughput value of 31.0–48.5 TPS. The throughput value of the query process is close to its send rate. However, the throughput for the invoking process includes create an item ID and transfer an item ID, is dissimilar. The throughput for the invoking process is about half of the send rates, ranging between 15.8 and 21.7 TPS. Although the throughput rate is low, but there is no failure in the results from capability tests.

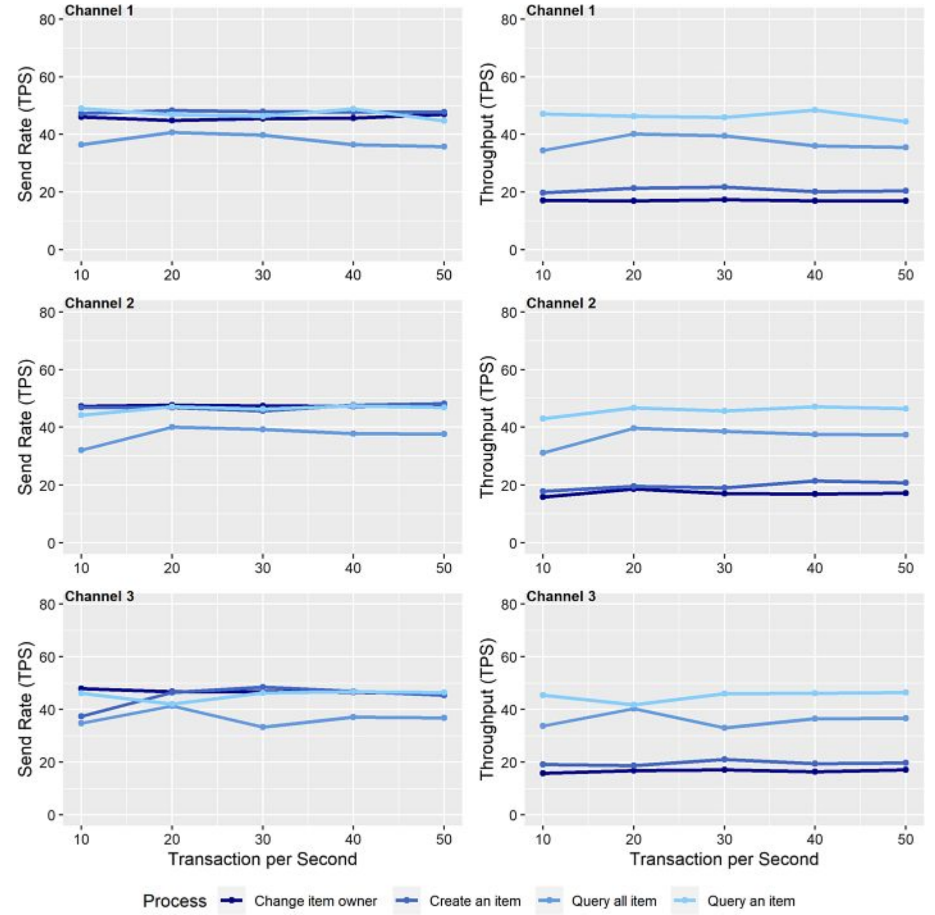


Fig. 6 Simulation result

From all the results obtained, it can be said that the blockchain model generates relatively constant transaction data per second regardless of the different speeds of each transaction. However, the throughput is disparate based on its process. The throughput of the query process is similar to its send rate and has a higher value than the invoking process. This is happened due process of the query only skim through the data in blockchain, while the invoking process is more difficult because the process involves creating new data in the blockchain.

Conclusions

Permissioned Blockchain is one of the newest technologies that are compatible with Halal Supply Chain, where administrators can determine the rights of each category of Halal Supply Chain participants, including what information is visible or what information can be added to the Blockchain. Usually, Halal Supply Chain participants such as suppliers, distributors, wholesalers, and retailers only focus on their respective parts. Thus, the role of regulators is also needed to determine the rights that exist (so that the smart contract on the blockchain can be used in accordance with the case). Of course, the determination of these rights must also be done by consensus so that no one feels disadvantaged.

From the web interface created, there are no failure in the validity test when the invoke test in the Create an Item ID and Transfer an Item ID process and when testing the query in the Query an Item ID and Query all Item ID processes. In addition, the web interface was also successfully tested for thwarting the formation of a block in case of data input errors from the user. In the integration between the web interface with the server, the server can do the process as a provider of information and validator for the web interface when the invoke and query process on the web interface is running or an input error occurs on the web interface that causes the failure of making blocks on the Blockchain.

Finally, from the results of simulations performed on the Blockchain Network created, as we can see, Blockchain's ability to secure transaction data is real because not all transaction processes fail. So, it is especially useful for securing transaction data about halal such as food, or drinks on the Blockchain. The "tampered-proof" capability also creates transparency for end users who want to examine the Halal Supply Chain process.

The implementation of blockchain architecture is important to improve the overall Halal Supply Chain. Blockchain could be executed without special requirements. Requirements for implementation of blockchain technology are personal computers that have Hyperledger architecture that is connected to a server.

In the future, the development of this research can be done using other Blockchain consensus methods such as Practical Byzantine Fault Tolerance and Zero Knowledge Proof. Other consensus methods can be simulated into the Blockchain Network system to see which Blockchain consensus performance could be improved. Some developments also can be done for Blockchain Network web interfaces, such as creating special programs to combine interfaces made with smartphones to be used more easily or develop user login systems through interfaces to validate Blockchain Network users. The interface of the Blockchain Network can also be synchronized with a barcode system to facilitate users who want to check the traceability of goods (Track and Trace). Furthermore, Artificial Intelligent can also be integrated into the Blockchain Network to make data input or data processing smooth.

Acknowledgements

Authors would like to express appreciation and gratitude to Universitas Indonesia for funding this study through PUTI Q2 Research Grants Universitas Indonesia No: NKB- 1702/UN2.RST/HKP.05.00/2020.

Authors' contributions

IS, HY, and EL designed the idea and draft, IS and HY conducted the programming and analyzed the result, EL and RM conducted the literature review and prepared the manuscript. All of the authors checked the final manuscript. All authors read and approved the final manuscript.

Funding

This Research is funded by Directorate of Research and Development Universitas Indonesia.

Availability of data and materials

All of data source is available on request from the authors.

Competing interests

All of the Authors declare that they do not have any particular competing interest.

Received: 4 September 2020 Accepted: 23 December 2020
Published online: 06 January 2021

References

1. Kehoe L, O'Connell N, Andrzejewski D, Gindner K, Dalal D. Introduction. In: When two chains combine: supply chain meets blockchain. Deloitte; 2017. https://www2.deloitte.com/content/dam/Deloitte/pt/Documents/blockchainsupplychain/IE_C_TL_Supplychain_meets_blockchain_.pdf. Accessed 20 Aug 2019.

2. Dinar Standard. Global Islamic economy key drivers. In: State of the global Islamic economy report 2019/20. Dubai International Financial Centre; 2019. https://salaamgateway.com/reports/report-state-of-the-global-islamic-economy-201920. Accessed 22 Aug 2019.
3. Grim BJ, Karim MS. Executive summary. In: The future of the global Muslim population: projections for 2010–2030. Pew Research Centre; 2011. http://assets.pewresearch.org/wp-content/uploads/sites/11/2011/01/FutureGlobalMuslimPopulation-WebPDF-Feb10.pdf. Accessed 26 Aug 2019.
4. Ngah AH, Zainuddin Y, Thurasamy R. Barriers and enablers in adopting of Halal warehousing. J Islam Mark. 2015;6(3):354–76. doi:https://doi.org/10.1108/JIMA-03-2014-0027.
5. Tieman M, Darun MR. Leveraging blockchain technology for halal supply chains. Islam Civ Renew. 2017;8(4):547–50. https://doi.org/10.12816/0045700.
6. Tieman M, Darun MR, Fernando Y, Ngah AB. Utilizing blockchain technology to enhance halal integrity: the perspectives of halal certification bodies. In: Xia Y, Zhang LJ, editors. Services—SERVICES 2019. CA: Academic; 2019. p. 119–28.
7. Simatupang TM Sistem Logistik Halal. Sistem Logistik Halal; 2016. http://supplychainindonesia.com. Accessed 20 Sept 2019.
8. Tieman M. The application of Halal in supply chain management: in-depth interviews. J Islam Mark. 2011;2(2):186–95. https://doi.org/10.1108/17590831111139893.
9. Khan S, Khan MI, Haleem A, Jami AR. Prioritising the risks in Halal food supply chain: an MCDM approach. J Islam Mark. 2019. https://doi.org/10.1108/JIMA-10-2018-0206.
10. Tieman M, van der Vorst JGAJ, Ghazali MC. Principles in halal supply chain management. J Islamic Mark. 2012;3(3):217–43. https://doi.org/10.1108/17590831211259727.
11. Ab Talib MS, Hamid ABA, Zulfakar MH. Halal supply chain critical success factors: a literature review. J Islam Mark. 2015;6(1):44–71. doi:https://doi.org/10.1108/JIMA-07-2013-0049.
12. Letourneau KB, Whelan ST, Blockchain. Staying ahead of tomorrow. J Equip Lease Fin. 2017;35(2):1–6.
13. Magista M, Dorra BL, Pean TY. A review of the applicability of gamification and game-based learning to improve household-level waste management practices among schoolchildren. Int J Tech. 2018;9(7):1439–49. doi:https://doi.org/10.14716/ijtechv9i7.2644.
14. Mavragani A, Tsagarakis KP. Predicting referendum results in the Big Data era. J Big Data. 2019. https://doi.org/10.1186/s40537-018-0166-z.
15. Hyperledger. Smart Contracts and Chaincode. Hyperledger Fabric; 2019. https://Hyperledger-Fabric.https://Hyperledger-Fabric.readthedocs.io/en/release-1.4/SmartContract/SmartContract.html. Accessed 20 Sept 2019.
16. Gupta M. Blockchain for dummies. 2nd ed. Hoboken: John Wiley & Sons, Inc; 2018.
17. Kshetri N. Blockchain's roles in meeting key supply chain management objectives. Int J Info Man. 2018;39:80–9. doi:https://doi.org/10.1016/j.ijinfomgt.2017.12.005.
18. Yusuf H, Surjandari I. Comparison of performance between kafka and raft as ordering service nodes implementation in hyperledger fabric. Int J Adv Sci Tech. 2020;29(7S):3549–54.
19. Yusuf H, Surjandari I, Rus AMM. Multiple channel with crash fault tolerant consensus blockchain network: a case study of vegetables supplier supply chain. In: Proceeding of the 2019 16th international conference on service systems and service management (ICSSSM); 2019 July 13–15; Shenzhen, China; 2019.
20. Sukhwani H. Hyperledger Fabric V1. In: Performance modeling & analysis of Hyperledger Fabric (Permissioned Blockchain Network). (Accessed 25 September 2019); https://dukespace.lib.duke.edu/dspace/bitstream/handle/10161/18268/Sukhwani_duke_0066D_14907.pdf?isAllowed=y&sequence=1 (Duke University, 2018).
21. Androulaki E, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the 13th ACM SIGOPS European conference on computer systems; 2018 April 23–26; Porto, Portugal; 2018.
22. Christidis K. A Kafka-based Ordering Service for Fabric. Google Docs; 2016. https://docs.google.com/document/d/19JihmW-8blTzN99lAubOfseLUZqdrB6sBR0HsRgCAnY/edit. Accessed 26 Aug 2019.
23. Hyperledger. The ordering service. Hyperledger Fabric. https://hyperledger-fabric.readthedocs.io/en/latest/orderer/ordering_service.html. Accessed 2 Sept 2019.
24. Alagappan R, Ganesan A, Liu J, Arpaci-Dusseau AC, Arpaci-Dusseau RH. (2018) Fault-tolerance, fast and slow: exploiting failure asynchrony in distributed systems. In: Proceedings of the 13th USENIX symposium on operating systems design and implementation (OSDI '18); 2018 October 8–10; Carlsbad, CA, USA; 2018.
25. Christidis K. Fabric Proposal: A raft-based ordering service. Google Docs; 2018. https://docs.google.com/document/d/138Brlx2BiYJm5bzFk_B0csuEUKYdXXr7Za9V7C76dwo/edit#. Accessed 15 Sept 2019.
26. Ongaro D, Ousterhout J. (2014) In search of an understandable consensus algorithm (extended version). In: Proceeding of USENIX annual technical conference, USENIX ATC 2014; June 19–20; Philadelphia, United States; 2014. p. 305–319.
27. Nguyen B, et al. Multichannel consensus. Google Docs; 2016. https://docs.google.com/document/d/1eRNxxQ0P8yp4Wh__Vi6ddaN_vhN2RQHP-IruHNUwyhc/edit. Accessed 15 Aug 2019.
28. OECD-FAO. Meat. In: OECD-FAO agricultural outlook 2019–2028. OECD Publishing, Paris/Food and Agriculture Organization of the United Nations;1999. http://www.fao.org/3/ca4076en/ca4076en.pdf. Accessed 30 Oct 2019.
29. Hyperledger. Blockchain Network. Hyperledger Fabric. https://hyperledger-fabric.readthedocs.io/en/release-1.4/network/network.html. Accessed 3 Sept 2019.
30. Geneiatakis D, Soupionis Y, Steri G, Kounelis I, Neisse R, Nai-Fovino I. Blockchain performance analysis for supporting cross-border E-Government services. IEEE Trans Eng Manag. 2020; 67(4): 1310–1322. https://doi.org/10.1109/TEM.2020.2979325

Publisher's note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.