J. Math. Computer Sci., 37 (2025), 347–360

Online: ISSN 2008-949X

Journal of Mathematics and Computer Science

Journal Homepage: www.isr-publications.com/jmcs

An inertial projective forward-backward-forward algorithm for constrained convex minimization problems and application to cardiovascular disease prediction

Prasit Cholamjiak[a,*], Watcharaporn Cholamjiak[a,*], Kunrada Kankam[b,*]

[a]School of Science, University of Phayao, Phayao 56000, Thailand.

[b]Elementary Education Program, Faculty of Education, Suan Dusit University Lampang Center, Lampang 52100, Thailand.

Abstract

In this paper, we introduce a novel machine learning algorithm designed for the classification of cardiovascular diseases. The proposed inertial projected forward-backward-forward algorithm is developed to address constrained minimization in Hilbert spaces, with a specific focus on improving the accuracy of disease classification. Utilizing inertial techniques, the algorithm employs a projected forward-backward-forward strategy, demonstrating convergence under mild conditions. Evaluation of the algorithm employs four essential performance metrics-accuracy, F1-score, recall, and precision to gauge its effectiveness compared to alternative classification models. Results indicate significant performance gains, achieving peak metrics of 77.50% accuracy, 71.57% precision, 91.27% recall, and 80.23% F1-score, thereby surpassing established benchmarks in machine learning models for cardiovascular disease classification.

Keywords: Projection method, inertial technique, classification problem, constrained minimization problem.

2020 MSC: 65K05, 90C25, 90C30.

1. Introduction and preliminaries

Various fields in applied sciences, engineering, computer science, medicine and etc. [2, 14, 18, 19, 27, 28, 33–36] can be formulated as mathematical model. Some problems have constraints that limit the set of possible solutions. To solve these problems, a common approach is to restrict changes within the working sub-space by adding or dropping one constraint at each iteration. In practical applications, many real-world problems such as image inpainting and data classification problems can be modeled as subproblems. To address these, a projected forward-backward algorithm can be employed to solve constrained convex minimization problems.

In this work, we aim to study the following constrained convex minimization problem:

$$\min_{x\in\Omega}(f(x)+g(x)) \tag{1.1}$$

*Corresponding author

Email addresses: prasit.ch@up.ac.th (Prasit Cholamjiak), watcharaporn.ch@up.ac.th (Watcharaporn Cholamjiak), kunradazzz@gmail.com (Kunrada Kankam)

doi: 10.22436/jmcs.037.03.08

Received: 2024-06-26 *Revised:* 2024-08-15 *Accepted:* 2024-09-11

where Ω is a nonempty closed and convex subset of a Hilbert space H and $f, g : H \to \mathbb{R} \cup \{+\infty\}$ are proper, convex, and lower semicontinuous functions that f is differentiable on H. The problem (1.1) relates to many real-world problems [9, 11, 17, 21, 29–31]. Let D be a convex subset of H, the strong relative inerior of D is

$$\text{sri } D = \{x \in D | \text{cone}(D - x) = \overline{\text{span}}(D - x)\}.$$

Let f and g be proper lower semicontinuous convex functions from H to $[-\infty, +\infty]$, and $\bar{x} \in H$. The subdifferential of g is denoted as the set-valued operator $\partial g : H \to 2^H$ and is defined as follows:

$$\partial g(x) = \{y \in H | g(w) - g(x) \geqslant \langle y, w - x\rangle, w \in H\}.$$

If $0 \in \text{sri}(\text{dom } f - \text{dom } g)$, then the following are equivalent (Corollary 26.3 in [4]):

i) $\bar{x}$ is a solution to the problem $\min_{x \in H}(f(x) + g(x))$;
ii) $\bar{x} \in \text{zer}(\partial f + \partial g) = \{x \in H | \ 0 \in (\partial f + \partial g)(x)\}$.

Moreover, if f is Gâteaux differentiable at $\bar{x}$, then above statements are equivalent to

$$\bar{x} = \text{prox}_{\beta g}(\bar{x} - \beta \nabla f(\bar{x})), \tag{1.2}$$

where $\beta > 0$ and ∇f is the gradient of f. If $\Omega = H$, then (1.1) solves the unconstrained convex minimization problem:

$$\min_{x \in H}(f(x) + g(x)). \tag{1.3}$$

For solving (1.3), we can construct a simple iteration. Let $x^0 \in H$ and

$$x^{n+1} = \text{prox}_{\beta g}(x^n - \beta \nabla f(x^n)), \tag{1.4}$$

where $\beta > 0$. By this point of view, we know that (1.4) is called a classical forward-backward splitting algorithm (FBS). As a consequence, it has been studied by many authors (see [6, 18–20, 28]). Let $x \in H$. We know that the orthogonal projection of x onto a nonempty, closed, and convex subset C of H is defined by

$$P_C x := \underset{y \in C}{\text{argmin}} \|x - y\|^2. \tag{1.5}$$

We know that

$$\|P_C x - y\|^2 \leqslant \|x - y\|^2 - \|P_C x - x\|^2,$$

for all $y \in C$. From (1.2) and (1.5), we can define a simple method for solving (1.1) as follows. Let $x^0 \in H$ and

$$x^{n+1} = P_\Omega(\text{prox}_{\beta g}(x^n - \beta \nabla f(x^n))), \tag{1.6}$$

where $\beta > 0$. The method (1.6) is called a projected forward-backward splitting algorithm (PFBS). In 2000, Tseng [32] introduced the forward-backward-forward splitting algorithm (FBFS) or Tseng's extragradient algorithm or Tseng's method. FBFS is generated by $x^0 \in H$ and

$$x^{n+1} = \text{prox}_{\beta_n g}(x^n - \beta_n \nabla f(x^n)) - \beta_n(\nabla f(\text{prox}_{\beta_n g}(x^n - \beta_n \nabla f(x^n))) - \nabla f(x^n)), \tag{1.7}$$

where (β_n) is a real positive sequence. In 2005, Combettes and Wajs [12] proposed the relaxed version of FBS (FBS-CW) which is generated by $x^0 \in H$, $\varepsilon \in (0, \min\{1, 1/L\})$ and

$$x^{n+1} = x^n + \lambda_n(\text{prox}_{\beta_n g}(x^n - \beta_n \nabla f(x^n)) - x^n), \tag{1.8}$$

where $\lambda_n \in [\varepsilon, 1]$, $\beta_n \in [\varepsilon, (2/L) - \varepsilon]$ and L is the Lipschitz constant of ∇f. In 2009, Beck and Teboulle [5] introduced a fast iterative shrinkage-thresholding algorithm (FISTA) as follows. Let $t_1 = 1$ and $x^0 = x^1 \in H$. Calculate $y^n = x^n + \theta_n(x^n - x^{n-1})$, where $\theta_n = \frac{t_n - 1}{t_{n+1}}$ and $t_{n+1} = \frac{1+\sqrt{1+4t_n^2}}{2}$. Next, calculate

$$x^{n+1} = \operatorname{prox}_{\alpha_n g}(y^n - \alpha_n \nabla f(x^n)), \tag{1.9}$$

where $\alpha_n = 1/L$ and L is the Lipschitz constant of ∇f. For other inertial methods, we refer to [3, 8, 10, 13, 22–24, 26].

In 2021, Phairatchatniyom et al. [25] introduced the modified inertial iterative algorithm (MII) for solving split variational inclusion problems in real Hilbert spaces as follows. Let $x^0 = x^1 \in H_1$ and set $\alpha > 0$, $\lambda > 0$ and $\{\gamma_n\} \subset [\gamma_*, \gamma^*] \subset (0, \frac{1}{L})$, where $L = \|A\|^2$. Let $\{\beta_n\} \subset (0,1)$ such that $\lim_{n\to\infty} \beta_n = 0$ and $\sum_{n=1}^{\infty} \beta_n = \infty$. Calculate $w_n = (1-\beta_n)[x_n + \theta_n(x_n - x_{n-1})]$, where $\{\theta_n\} \subset [0,1)$ such that

$$\theta_n = \begin{cases} \min\left\{\dfrac{\varepsilon_n}{\|x_n - x_{n-1}\|}, \theta\right\}, & \text{if } x_n \neq x_{n-1}, \\ \theta, & \text{otherwise.} \end{cases}$$

Next, calculate

$$x^{n+1} = J_\lambda^{B_1}(w_n + \alpha_n A^*(J_\lambda^{B_2} - I)Aw_n). \tag{1.10}$$

In 2022, Adamu et al. [1] introduced inertial Halpern-type forward-backward splitting algorithm (IHFB) for solving variational inclusion problems in a real Banach space E as follows. Let $x^0 = x^1 \in E$ and choose θ_n such that $0 \leqslant \theta_n \leqslant \bar{\theta}_n$, where

$$\bar{\theta}_n = \begin{cases} \min\left\{\dfrac{\varepsilon_n}{\|x_n - x_{n-1}\|}, \theta\right\}, & \text{if } x_n \neq x_{n-1}, \\ \theta, & \text{otherwise.} \end{cases}$$

Calculate $y_n = x_n + \theta_n(x_n - x_{n-1})$. Next, compute $v_n = \beta_n u + (1-\beta_n)J_{\lambda_n}^B(y_n - \lambda_n A y_n)$ and

$$x_{n+1} = \eta_n y_n + (1-\eta_n)v_n. \tag{1.11}$$

Highlights for this research are following.

- This research introduces new projected forward-backward-forward algorithms based on Ishikawa iterations, designed to address constrained convex minimization problems. Using employing the inertial technique, the study provides a weak convergence theorem for the proposed algorithm.
- The algorithm is applied to data classification problems, specifically in Section 3 we present a proposed algorithm to predict cardiovascular disease datasets.
- The proposed algorithm demonstrates good performance compared to existing methods, with enhanced precision, recall, F1 score, and accuracy. It effectively learns from training datasets and generalizes well to holdout datasets, outperforming other methods in the literature. With more training models, the results show that the loss tends to decrease, and the accuracy tends to increase, this means that our Algorithms suitably learn the training dataset in machine learning.

The content is arranged as follows. In Section 2, we construct our main theorem. In Section 3, we test experiments and discuss applications for our algorithms. Finally, in Section 4, we end this work by conclusions.

2. Main theorem

In this section, we suggest a new inertial projected forward-backward-forward splitting algorithm and establish the weak convergence. Let Ω be a nonempty closed and convex subset of H. Now, we assume that $f : H \to \mathbb{R} \cup \{+\infty\}$ and $g : H \to \mathbb{R} \cup \{+\infty\}$ are proper, lower semi-continuous, and convex functions that f is differentiable on H with the Lipschitz constant L of ∇f. Throughout this paper, we assume that $\Omega \cap \operatorname{argmin}(f+g)$ is nonempty.

Algorithm 2.1 (Inertial projected forward-backward-forward splitting algorithm (IPFBF)**).**
Initialization: Given $\theta_n \in [0, +\infty)$, $\eta_n \in (0, 2/L)$ and $\alpha_n \in (0, 1/L)$.
Iterative step: Let $x^0, x^1 \in H$ and calculate x^{n+1} as follows:
Step1. Compute the inertial step: $y^n = x^n + \theta_n(x^n - x^{n-1})$.
Step2. Compute the forward-backward step:

$$z^n = \operatorname{prox}_{\eta_n g}(y^n - \eta_n \nabla f(y^n)) \quad \text{and} \quad w^n = \operatorname{prox}_{\alpha_n g}(z^n - \alpha_n \nabla f(z^n)).$$

Step3. Compute the x^{n+1} step: $x^{n+1} = P_\Omega(w^n + \alpha_n(\nabla f(z^n) - \nabla f(w^n)))$.
Set $n := n+1$ and return to **Step1.**

Theorem 2.2. *Let (x^n) be generated by Algorithm 2.1 and $\theta_n \in [0, \infty)$. Assume that $0 < \liminf_{n\to\infty} \eta_n \leqslant \limsup_{n\to\infty} \eta_n < \frac{2}{L}$, $0 < \liminf_{n\to\infty} \alpha_n \leqslant \limsup_{n\to\infty} \alpha_n < \frac{1}{L}$, and $\sum_{n=1}^{\infty} \theta_n < +\infty$. Then we have*

1. *for each $x_* \in \Omega \cap \operatorname{argmin}(f+g)$, $\|x^{n+1} - x_*\| \leqslant K \cdot \prod_{j=1}^{n}(1 + 2\theta_j)$, where $K = \max\{\|x^0 - x_*\|, \|x^1 - x_*\|\}$;*
2. *(x^n) weakly converges to an element of $\Omega \cap \operatorname{argmin}(f+g)$.*

Proof. Let $x_* \in \Omega \cap \operatorname{argmin}(f+g)$ and $q^n = w^n + \alpha_n(\nabla f(z^n) - \nabla f(w^n))$. So, we get

$$\|x^{n+1} - x_*\|^2 = \|P_\Omega(q^n) - x_*\|^2 \leqslant \|q^n - x_*\|^2 - \|P_\Omega(q^n) - q^n\|^2. \tag{2.1}$$

By the definition of w^n, we have

$$z^n - w^n - \alpha_n \nabla f(z^n) \in \alpha_n \partial g(w^n). \tag{2.2}$$

Moreover, we see that

$$\alpha_n \nabla f(z^n) = q^n - w^n + \alpha_n \nabla f(w^n). \tag{2.3}$$

Using (2.2) and (2.3), we obtain

$$z^n - q^n - \alpha_n \nabla f(w^n) \in \alpha_n \partial g(w^n). \tag{2.4}$$

Noting $x_* \in \operatorname{argmin}(f+g)$, we get $-\alpha_n \nabla f(x_*) \in \alpha_n \partial g(x_*)$. Using (2.4) and the monotonicity of ∂g, it gives

$$\langle z^n - q^n - \alpha_n(\nabla f(w^n) - \nabla f(x_*)), w^n - x_* \rangle \geqslant 0. \tag{2.5}$$

Since ∇f is monotone, it follows from (2.5) that $\langle z^n - q^n, w^n - x_* \rangle \geqslant 0$. Therefore, we get

$$\langle z^n - q^n, w^n - q^n \rangle + \langle z^n - q^n, q^n - x_* \rangle \geqslant 0. \tag{2.6}$$

We know that $\|a \pm b\|^2 = \|a\|^2 \pm 2\langle a, b \rangle + \|b\|^2$ for all $a, b \in H$. From (2.6), we have

$$\frac{1}{2}\left[\|z^n - q^n\|^2 + \|q^n - w^n\|^2 - \|z^n - w^n\|^2\right] + \frac{1}{2}\left[\|z^n - x_*\|^2 - \|z^n - q^n\|^2 - \|q^n - x_*\|^2\right] \geqslant 0.$$

This implies that

$$\|q^n - x_*\|^2 \leqslant \|z^n - x_*\|^2 + \|q^n - w^n\|^2 - \|z^n - w^n\|^2. \tag{2.7}$$

Since ∇f is L-Lipschitz continuous on H, we have

$$\|q^n - w^n\|^2 = \|w^n + \alpha_n(\nabla f(z^n) - \nabla f(w^n)) - w^n\|^2 \leqslant \alpha_n^2 L^2 \|z^n - w^n\|^2. \tag{2.8}$$

Moreover, we see that

$$\begin{aligned}
\|z^n - x_*\|^2 &= \|\mathrm{prox}_{\eta_n g}(y^n - \eta_n \nabla f(y^n)) - \mathrm{prox}_{\eta_n g}(x_* - \eta_n \nabla f(x_*))\|^2 \\
&\leqslant \|(y^n - \eta_n \nabla f(y^n)) - (x_* - \eta_n \nabla f(x_*))\|^2 \\
&\quad - \|(I - \mathrm{prox}_{\eta_n g})(y^n - \eta_n \nabla f(y^n)) - (I - \mathrm{prox}_{\eta_n g})(x_* - \eta_n \nabla f(x_*))\|^2 \\
&\leqslant \|y^n - x_*\|^2 - \frac{2\eta_n}{L}\|\nabla f(y^n) - \nabla f(x_*)\|^2 + \eta_n^2 \|\nabla f(y^n) - \nabla f(x_*)\|^2 \\
&\quad - \|y^n - \mathrm{prox}_{\eta_n g}(y^n - \eta_n \nabla f(y^n)) - \eta_n(\nabla f(y^n) - \nabla f(x_*))\|^2 \\
&= \|y^n - x_*\|^2 - \eta_n(\frac{2}{L} - \eta_n)\|\nabla f(y^n) - \nabla f(x_*)\|^2 \\
&\quad - \|y^n - \mathrm{prox}_{\eta_n g}(y^n - \eta_n \nabla f(y^n)) - \eta_n(\nabla f(y^n) - \nabla f(x_*))\|^2.
\end{aligned} \tag{2.9}$$

From (2.7), (2.8), and (2.9), we obtain

$$\begin{aligned}
\|q^n - x_*\|^2 &\leqslant \|z^n - x_*\|^2 + \alpha_n^2 L^2 \|z^n - w^n\|^2 - \|z^n - w^n\|^2 \\
&= \|z^n - x_*\|^2 - (1 - \alpha_n^2 L^2)\|z^n - w^n\|^2 \\
&\leqslant \|y^n - x_*\|^2 - \eta_n(\frac{2}{L} - \eta_n)\|\nabla f(y^n) - \nabla f(x_*)\|^2 - \|y^n - \mathrm{prox}_{\eta_n g}(y^n - \eta_n \nabla f(y^n)) \\
&\quad - \eta_n(\nabla f(y^n) - \nabla f(x_*))\|^2 - (1 - \alpha_n^2 L^2)\|z^n - w^n\|^2.
\end{aligned} \tag{2.10}$$

Combining (2.1) and (2.10), we get

$$\begin{aligned}
\|x^{n+1} - x_*\|^2 \leqslant \|y^n - x_*\|^2 &- \eta_n(\frac{2}{L} - \eta_n)\|\nabla f(y^n) - \nabla f(x_*)\|^2 \\
&- \|y^n - \mathrm{prox}_{\eta_n g}(y^n - \eta_n \nabla f(y^n)) - \eta_n(\nabla f(y^n) - \nabla f(x_*))\|^2 \\
&- (1 - \alpha_n^2 L^2)\|z^n - w^n\|^2 - \|P_\Omega(q^n) - q^n\|^2.
\end{aligned} \tag{2.11}$$

Now, we will show that (x^n) is bounded. From definition of y^n and (2.11), we see that

$$\|x^{n+1} - x_*\| \leqslant \|y^n - x_*\| = \|x^n + \theta_n(x^n - x^{n-1}) - x_*\| \leqslant \|x^n - x_*\| + \theta_n(\|x^n - x_*\| + \|x^{n-1} - x_*\|).$$

This shows that

$$\|x^{n+1} - x_*\| \leqslant (1 + \theta_n)\|x^n - x_*\| + \theta_n \|x^{n-1} - x_*\|.$$

By Lemma 5 in [14], we conclude that

$$\|x^{n+1} - x_*\| \leqslant K \cdot \prod_{j=1}^{n}(1 + 2\theta_j),$$

where $K = \max\{\|x^0 - x_*\|, \|x^1 - x_*\|\}$. Since $\sum_{n=1}^{\infty} \theta_n < +\infty$, we have (x^n) is bounded. By (2.11), we obtain

$$\begin{aligned}
\|x^{n+1}-x_*\|^2 &\leqslant \|x^n+\theta_n(x^n-x^{n-1})-x_*\|^2-\eta_n(\frac{2}{L}-\eta_n)\|\nabla f(y^n)-\nabla f(x_*)\|^2\\
&\quad -\|y^n-\mathrm{prox}_{\eta_n g}(y^n-\eta_n\nabla f(y^n))-\eta_n(\nabla f(y^n)-\nabla f(x_*))\|^2\\
&\quad -(1-\alpha_n^2L^2)\|z^n-w^n\|^2-\|P_\Omega(q^n)-q^n\|^2\\
&\leqslant \|x^n-x_*\|^2+2\theta_n\|x^n-x_*\|\|x^n-x^{n-1}\|+\theta_n^2\|x^n-x^{n-1}\|^2\\
&\quad \nu-\eta_n(\frac{2}{L}-\eta_n)\|\nabla f(y^n)-\nabla f(x_*)\|^2-\|y^n-\mathrm{prox}_{\eta_n g}(y^n-\eta_n\nabla f(y^n))\\
&\quad -\eta_n(\nabla f(y^n)-\nabla f(x_*))\|^2-(1-\alpha_n^2L^2)\|z^n-w^n\|^2-\|P_\Omega(q^n)-q^n\|^2.
\end{aligned}$$

Since $\lim_{n\to\infty}\theta_n\|x^n-x^{n-1}\|=0$, $\lim_{n\to\infty}\|x^n-x_*\|$ exists and $0<\liminf_{n\to\infty}\eta_n\leqslant\limsup_{n\to\infty}\eta_n<\frac{2}{L}$, we have

$$\lim_{n\to\infty}\|\nabla f(y^n)-\nabla f(x_*)\|=0$$

and

$$\lim_{n\to\infty}\|y^n-\mathrm{prox}_{\eta_n g}(y^n-\eta_n\nabla f(y^n))-\eta_n(\nabla f(y^n)-\nabla f(x_*))\|=0.$$

So $\lim_{n\to\infty}\|y^n-z^n\|=0$. Moreover, we can show that $\lim_{n\to\infty}\|P_\Omega(q^n)-q^n\|=0$ and $\lim_{n\to\infty}\|z^n-w^n\|=0$. By definition of y^n, it is easily seen that $\lim_{n\to\infty}\|x^n-y^n\|=0$. Then,

$$\|z^n-x^n\|\leqslant\|z^n-y^n\|+\|y^n-x^n\|\rightharpoonup 0 \text{ as } n\to\infty.$$

Moreover, we see that

$$\begin{aligned}
\|q^n-x^n\| &\leqslant \|q^n-w^n\|+\|w^n-z^n\|+\|z^n-x^n\|\\
&\leqslant \alpha_n^2L^2\|w^n-z^n\|+\|w^n-z^n\|+\|z^n-x^n\|\\
&=(1+\alpha_n^2L^2)\|w^n-z^n\|+\|z^n-x^n\|\to 0 \text{ as } n\to\infty.
\end{aligned}$$

By definition of z^n, we obtain

$$\frac{y^n-\eta_n\nabla f(y^n)-z^n}{\eta_n}\in\partial g(z^n)$$

that is

$$\frac{y^n-z^n}{\eta_n}-\nabla f(y^n)+\nabla f(z^n)\in\partial g(z^n)+\nabla f(z^n). \tag{2.12}$$

Therefore

$$\begin{aligned}
\left\|\frac{y^n-z^n}{\eta_n}-\nabla f(y^n)+\nabla f(z^n)\right\| &\leqslant \frac{1}{\eta_n}\|y^n-z^n\|+\|\nabla f(y^n)-\nabla f(z^n)\|\\
&\leqslant\left(\frac{1}{\eta_n}+L\right)\|y^n-z^n\|\to 0 \text{ as } n\to\infty.
\end{aligned}$$

If x^∞ is a weak limit point of (x^n), then there exists a subsequence (x^{n_i}) of (x^n) such that $x^{n_i}\rightharpoonup x^\infty$. By replacing n by n_i and passing $i\to\infty$ in (2.12), we have $0\in\partial g(x^\infty)+\nabla f(x^\infty)$ by Fact 2.2 in [6]. On the other hand since $\lim_{n\to\infty}\|P_\Omega(q^n)-q^n\|=0$ and $q^{n_i}\rightharpoonup x^\infty$, by the demiclosedness of P_Ω, we have $x^\infty\in\Omega$. Therefore $x^\infty\in\Omega\cap\mathrm{argmin}(f+g)$. Using Theorem 5.5 in [4], Theorem 2.2 is completed. □

3. Application to data classification

Many real world problems such as signal and image processing, transportation, data regression, and classification problems can be formulated in the form of a convex minimization problem. In this section, we focus on demonstrating how to format some machine learning (ML) problems, especially a classification problem, into a constrained convex minimization problems by using the proposed algorithm. Also, it shows that our algorithms are more effective than some algorithms mentioned in the literature. Precisely, we are interested in cardiovascular disease (CVD) for the dataset. It was obtained from the Kaggle dataset (https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset) for cardiovascular disease prediction, an online resource.

Cardiovascular disease is one reason for death in the world, and every year, around 17.9 million deaths, representing 32% of all global deaths. Of these deaths, 85% were due to heart attack and stroke are estimated by World Health Organization. At least three-quarters of the world's deaths from CVD occur in low- and middle-income countries. People living in low- and middle-income countries often do not have the benefit of primary health care programmes for early detection and treatment of people with risk factors for CVD. The most important behavioural risk factors of heart disease and stroke are unhealthy diet, physical inactivity, tobacco use, and harmful use of alcohol. The effects of behavioural risk factors may show up in individuals. We use the CVD dataset in Kaggle to predict cardiovascular disease in response to the situation.

The dataset consists of 70,000 records and 12 characteristics. There are two categories of analyzing the classes: 0 for absence or 1 for the presence of the disease. From the data samples, 35021 (50.03%) are specified as absent, while the remaining 34,979 (49.97%) are identified as the presence of cardiovascular disease. For benchmarking classifier, we consider the following various methods which have been proposed to classify CVD as in Table 1.

Table 1: Classification accuracy of different methods.

Methods	Authors	Acc (%)
Machine Learning Techniques	Jinjri et al. [16]	
Logistic Regression		69.87
K-Nearest Neighbour		72.36
Support Vector Machine		72.66
Machine Learning Techniques	Bhave and Gaikwad [7]	
Decision Trees		64.22
Random forest		72.08
AdaBoost		73.28
XGBoot		74.11
LightGBM		74.07
Majority viting ensemble		74.17
ELM-Algorithm 2.1		**77.50**

The CVD dataset is considered relevant for identifying the disease. The goal column is divided into two categories: 1 for heart problems and 0 for disorders other than heart disease. The attributes were collected during the patient's medical examination: 1. objective data (factual information); 2. examination data (results from medical exams); 3. subjective data (information given by the patient). Table 2 contains a detailed description of the characteristics.

Table 2: Distribution of features of the study population.

Attribute name	**Description**	**Value type**	**Variable type**
Age	Patient's age in days	int (days)	Objective
Height	Patient's height in cm	int (cm)	
Weight	Patient's weight in kg	float (kg)	
Gender	Patient's gender	(1) women (2) men	
Ap–hi	Systolic blood pressure	int	Examination
Ap–lo	Diastolic blood pressure	int	
Cholesterol	Patient's cholesterol	(1) normal (2) above normal (3) well above normal	
Gluc	Patient's glucose	(1) normal (2) above normal (3) well above normal	
Smoke	Whether patient smokes or not	binary	Subjective
Alco	Whether the patient consumes alcohol or not	binary	
Active	Whether patient is physically active or not	binary	
Cardio	Presence or absence of cardiovascular disease	binary	Target

Let $S := \{(y^n, b^n) | y^n \in \mathbb{R}^q, b^n \in \mathbb{R}^p, n = 1, 2, \dots, J\}$ be the training dataset, where J is distinct samples, y^n is an input data, and b^n is a target. In experiments on regression and classification problems, the main goal of extreme learning machine (ELM) is to find $x = [{x^1}^T, \dots, {x^M}^T]^T$ such that $Ax = b$, where M is the number of nodes in the hidden layer and A is hidden layer output matrix defied by

$$A = \begin{bmatrix} G(a^1 y^1 + b^1) & \cdots & G(a^M y^1 + b^M) \\ \vdots & \ddots & \vdots \\ G(a^1 y^K + b^1) & \cdots & G(a^M y^K + b^M) \end{bmatrix},$$

G ia an activate function, a^i and b^i are random weight and bias of the i-th hidden node, and $b = [{b^1}^T, \dots, {b^J}^T]^T$ is the training data. The output at the i-th hidden node is

$$O^n = \sum_{i=1}^{M} x^i G(a^i y^n + b^i).$$

The classification problems can be expressed as Table 3.

To predict CVD, we apply the proposed algorithm to solve all models in Table 3. All results are performed by MATLAB 2022b on a 64-bit MacBook Pro Chip Apple M1 and 8 GB of RAM. To evaluate the quality of the predicted dataset, we use precision, recall, F1 score, and accuracy, which are defined by

$$\text{precision} = \frac{TP}{TP + FP}, \quad \text{recall} = \frac{TP}{TP + FN}, \quad \text{F1 score} = \frac{2TP}{2TP + FP + FN},$$

and

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%,$$

Table 3: List of models considered for the proposed algorithm, where λ, λ_1, and λ_2 are a regularization parameter.

Models	Function	Explanation	Argument
M1	$\mathrm{prox}_{\ell 1}$	ℓ_1 norm proximal operator. Solve: $\min_{x \in H} \|Ax - b\|_2^2 + \lambda \|x\|_1$	x, λ, parameters
The problem (1.3) setting: $f(x) = \|Ax - b\|_2^2$ and $g(x) = \lambda \|x\|_1$.			
M2	$\mathrm{prox}_{\ell 2}$	ℓ_2 norm proximal operator. Solve: $\min_{x \in H} \|Ax - b\|_2^2 + \lambda \|x\|_2^2$	x, λ, parameters
The problem (1.3) setting: $f(x) = \|Ax - b\|_2^2$ and $g(x) = \lambda \|x\|_2^2$.			
M3	$\mathrm{prox}_{\ell 1}$	ℓ_1 norm proximal operator. Solve: $\min_{x \in C} \|Ax - b\|_2^2 + \lambda_1 \|x\|_1$	x, λ_1, λ_2, parameters
The problem (1.1) setting: $f(x) = \|Ax - b\|_2^2$, $g(x) = \lambda \|x\|_1$, and $C = \{x \in \mathbb{R} \mid \|x\|_1 < \lambda_2\}$.			
M4	$\mathrm{prox}_{\ell 2}$	ℓ_2 norm proximal operator. Solve: $\min_{x \in C} \|Ax - b\|_2^2 + \lambda_1 \|x\|_2^2$	x, λ_1, λ_2, parameters
The problem (1.1) setting: $f(x) = \|Ax - b\|_2^2$, $g(x) = \lambda \|x\|_2^2$, and $C = \{x \in \mathbb{R} \mid \|x\|_2^2 < \lambda_2\}$.			

where TP is a true positive, TN is a true negative, FP is a false positive, and FN is false negative. The binary cross entropy loss function calculates the loss of an example by computing the following average:

$$\text{Loss} = -\frac{1}{\text{output size}} \sum_{i=1}^{\text{output size}} y_i \log \bar{y}_i + (1 - y_i) \log(1 - \bar{y}_i),$$

where output size is the number of scalar values in the model output, y_i is a corresponding target value, and $\bar{y}_i$ is a i-th scalar value in the model output.

To start our computation, we clean the dataset by apps in MATLAB using smooth data for attributes: height, weight, Ap–hi, and Ap–lo as in Tables 4 and 5. 70% of the dataset was selected as the training set and 30% as the test set to cross-validate the model's performance, and adjust the classification model according to the parameters of the classification algorithm.

Table 4: Setting method and parameters for cleaned dataset.

Specify method and parameters	**Attribute names**			
	Height	Weight	Systolic blood pressure	Diastolic blood pressure
Smoothing method	Moving mean	Moving mean	Moving median	Moving median
Smoothing parameters	Smoothing factor	Smoothing factor	Smoothing factor	Smoothing factor
Smoothing factor	0.25	0.3	0.65	0.85

The initial points $x^0 = x^1$ are zero vectors with the size of training dataset for all algorithms. The parameters $\eta_n = \frac{0.9}{2\max(\mathrm{eig}(A^T A))}$ and $\alpha_n = \frac{0.9}{\max(\mathrm{eig}(A^T A))}$, where $\mathrm{eig}(A^T A)$ is eigenvalues of $A^T A$. θ_n of all algorithms is defined as

$$\theta_n = \begin{cases} \frac{t_n - 1}{t_{n+1}}, \text{ where } t_{n+1} = \frac{1 + \sqrt{1 + 4t_n^2}}{2}, & \text{if } 1 \leqslant n \leqslant M, \\ \frac{1}{n^2}, & \text{otherwise,} \end{cases}$$

Table 5: Results of the cleaned dataset.

Attribute names		Maximum (Minimum)	Mean	Standard Deviation (SD)
Height	Original	250 (55)	164.36	8.21
	Cleaned	209 (105.5)	164.36	5.79
Weight	Original	200 (10)	74.21	14.40
	Cleaned	144.5 (37)	74.21	10.34
Systolic blood pressure (Ap_hi)	Original	16020 (-150)	128.82	154.01
	Cleaned	210 (80)	126.13	11.95
Diastolic blood pressure (Ap_lo)	Original	11000 (-70)	96.63	188.47
	Cleaned	110 (60)	81.60	4.95

for some positive integer M. The sigmoid is an activation function, hidden nodes $M = 110$, and the binary cross entropy loss = 0.15 for the stopping criteria. The results as shown in Table 6.

Table 6: The performance for solving models in Table 3.

Models	Regularization parameters	ITER (CPU)	Pre	Rec	F1	Acc(%)
M1	$\lambda = 0.01$	48 (2.1696)	71.5789	91.2621	80.2310	**77.5000**
M2	$\lambda = 10^{-5}$	48 (2.2437)	71.5757	91.2717	80.2326	**77.5000**
M3	$\lambda_1 = 10^{-5}$ and $\lambda_2 = 0.5$	51 (2.3678)	71.5041	91.4525	80.2573	77.4905
M4	$\lambda_1 = 10^{-5}$ and $\lambda_2 = 0.5$	51 (2.4462)	71.5009	91.4620	80.2589	77.4905

From Table 6, we see that model M1 performs better than models M3 and M4 in terms of quality of the predicted dataset, number of iterations and CPU time. However, model M1 has an accuracy value equal to M2 (77.5%), but in terms of CPU time, model M1 performs better than M2.

Next, we apply different ML algorithms such as FBFS (1.7), FBS-CW (1.8), FISTA (1.9), MII (1.10), and IHFB (1.11) to solve Model M1 and compare the efficiency of algorithms. All parameters are chosen as in Table 7. The initial points $x^0 = x^1$ are zero vectors with the size of training dataset for all algorithms. θ_n is defined by

$$\theta_n = \begin{cases} \dfrac{t_n - 1}{t_{n+1}}, \text{ where } t_{n+1} = \dfrac{1+\sqrt{1+4t_n^2}}{2}, & \text{if } 1 \leqslant n \leqslant M, \\ \dfrac{1}{n^2}, & \text{otherwise}, \end{cases}$$

for FBFS FBS-CW FISTA and IPFB. We choose θ_n by

$$\theta_n = \begin{cases} \min\left\{ \dfrac{\varepsilon_n}{\|x_n - x_{n-1}\|}, \theta \right\}, & \text{if } x_n \neq x_{n-1}, \\ \theta, & \text{otherwise}, \end{cases}$$

for IHFB and MII. The sigmoid is an activation function, hidden nodes $M = 110$, and the binary cross entropy loss = 0.15 for the stopping criteria. The results are shown in Table 8.

From Table 8, we see that the IPFBF algorithm has an excellent fit model. It suitably learns the training dataset and generalizes to classify the CVD dataset better than FBFS (1.7), FBS-CW (1.8), FISTA (1.9), MII (1.10), and IHFB (1.11) for these parameters. Moreover, The performance of the IPFBF algorithm has the highest efficiency in precision, recall, F1 score, and accuracy in the case of the same number of iterations. Confusion matrix for IPFBF algorithm is shown in Figure 1. Also, we present the accuracy value to show efficiency of our algorithm and the validation loss with the accuracy of training. This shows that it has no overfitting in the training dataset in Figures 2 and 3, respectively.

Table 7: Chosen parameters of each algorithm.

Methods	Parameters							
	α_n	η_n	β_n	λ_n	t_1	ε_n	θ	u
FBFS	–	–	$\frac{0.1}{\max(\mathrm{eig}(A^TA))}$	–	1	–	–	–
FBS-CW	–	–	$\frac{0.1}{\max(\mathrm{eig}(A^TA))}$	$\frac{0.1}{\max(\mathrm{eig}(A^TA))}$	1	–	–	–
FISTA	$\frac{0.9}{\max(\mathrm{eig}(A^TA))}$	–	–	–	1	–	–	–
IHFB	–	$\frac{1}{(n+1)^8}$	$\frac{1}{1000n}$	$\frac{0.1}{\max(\mathrm{eig}(A^TA))}$	–	$\frac{1}{(n+1)^6}$	0.8	$\frac{b}{2}$
MII	$\frac{0.9}{\max(\mathrm{eig}(A^TA))}$	–	$\frac{1}{(n+2)^{1/2}}$	–	–	$\left(\frac{0.1}{\max(\mathrm{eig}(A^TA))}\right)^3$	0.01	–
IPFB	$\frac{0.9}{\max(\mathrm{eig}(A^TA))}$	$\frac{0.9}{\max(\mathrm{eig}(A^TA))}$	–	–	1	–	–	–

Table 8: The performance for solving model M1.

Algorithms	**Regularization parameters**	**ITER (CPU)**	**Pre**	**Rec**	**F1**	**Acc(%)**
FBFS	$\lambda = 10^{-5}$	48 (1.6328)	69.3099	93.0135	79.4310	75.9000
FBS-CW	$\lambda = 10^{-5}$	48 (1.0998)	69.6199	92.7470	79.5364	76.1238
FISTA	$\lambda = 10^{-5}$	48 (0.5523)	70.6676	92.1854	80.0051	76.9476
IHFB	$\lambda = 10^{-5}$	48 (0.6561)	69.3099	93.0135	79.4310	75.9000
MII	$\lambda = 10^{-2}$	48 (0.6478)	64.9894	96.2593	77.5924	72.1857
IPFBF	$\lambda = 10^{-5}$	48 (2.2437)	**71.5757**	**91.2717**	**80.2326**	**77.5000**

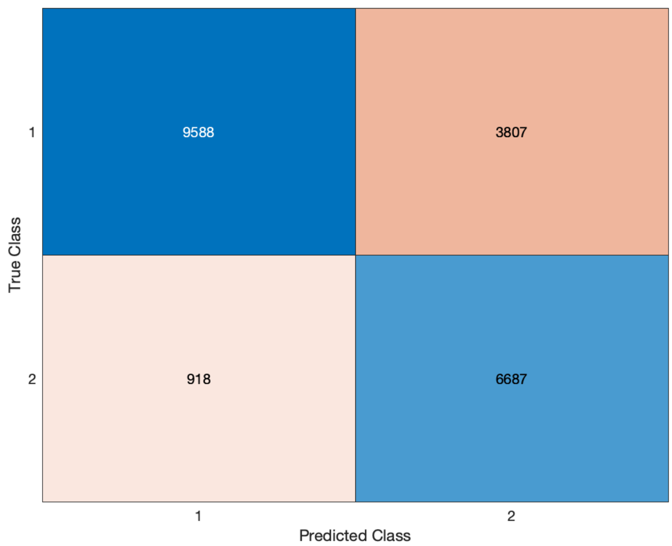

Figure 1: Confusion matrix for IPFBF algorithm.

Figure 1 shows that the confusion matrix gives a true positive (TP: 9588), false positive (FP: 3807), false negative (FN: 918), and true negative (TN: 6687). Although satisfactory, the results could be better to be implemented in clinical settings where excellent performance is required given the much FN and FP. However, it can be used to predict CVD for preliminary checks in the future.

From Figures 2 and 3, we see that the IPFBF algorithm has an excellent fit model. It suitably learns the training dataset and generalizes to classify the CVD dataset.

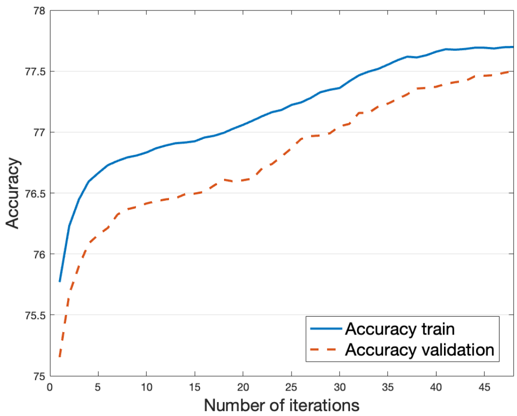

Figure 2: Performance graph of IPFBF algorithm (accuracy train: 77.6980%, accuracy validation: 77.5000%) on CVD dataset.

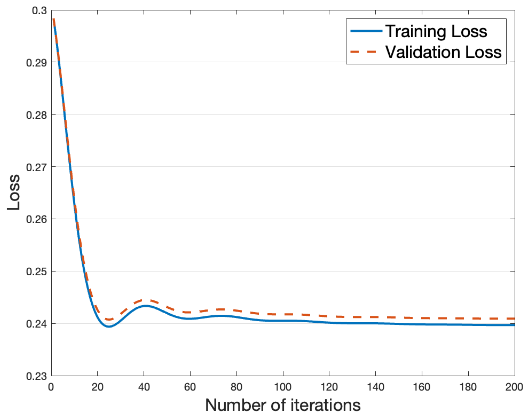

Figure 3: Loss curve of IPFBF algorithm (training loss: 0.2397%, validation loss: 0.2409%) on CVD dataset.

4. Conclusion

In this work, we introduced an inertial projected forward-backwards-forward splitting algorithm for solving the constrained convex minimization problem in Hilbert spaces. Under the suitable conditions, we then proved that the proposed algorithm converges weakly to the solution. In the numerical experiments, we applied the proposed algorithm to solve the CVD classification in infinite dimensional spaces and compared the performance of the algorithm with FBFS (1.7), FBS-CW (1.8), and FISTA (1.9). It is shown that the classification efficiency varies with the selected method. The results show that our algorithms are effective regarding precision, recall, F1 score, and accuracy. Based on the experiments, the proposed algorithm performs better than other methods in terms of accuracy at 77.50%. It shows that our algorithm suitably learns the training dataset and generalizes well to classify.

Data availability

The cardiovascular disease dataset was obtained from the Kaggle dataset (https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset).

Author contributions

Supervision and reviewing, P.C.; Reviewing and editing, W.C.; Writing manuscript and software computing, K.K.

Acknowledgments

This research was supported by University of Phayao and Thailand Science Research and Innovation Fund (Fundamental Fund 2025) and School of Science, University of Phayao, under grant No. PBTSC67037.

References

[1] A. Adamu, D. Kitkuan, T. Seangwattana, *An accelerated Halpern-type algorithm for solving variational inclusion problems with applications*, Bangmod Int. J. Math. Comput. Sci., **8** (2022), 37–55. 1

[2] M. Ariyanatchi, G. M. Vijayalakshmi, *Nonlinear Robust Adaptive Sliding Mode Control Strategies Involve a Fractional Ordered Approach to Reducing Dengue Vectors*, Results Control Optim., **14** (2024), 13 pages. 1

[3] H. Attouch, J. Peypouquet, *The rate of convergence of Nesterov's accelerated forward-backward method is actually faster than* $1/k^2$, SIAM J. Optim., **26** (2016), 1824–1834. 1

[4] H. H. Bauschke, P. L. Combettes *Convex analysis and monotone operator theory in Hilbert spaces*, Springer, New York, (2011). 1, 2
[5] A. Beck, M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., **2** (2009), 183–202. 1
[6] J. Y. Bello Cruz, T. T. A. Nghia, *On the convergence of the forward-backward splitting method with linesearches*, Optim. Methods Softw., **31** (2016), 1209–1238. 1, 2
[7] A. Bhave, S. Gaikwad, *Prediction of Cardiovascular Disease using Machine Learning Algorithms and Ensemble Techniques*, Int. J. Comput. Sci. Commun., **13** (2021), 14–23. 1
[8] W. Cholamjiak, P. Cholamjiak, S. Suantai, *An inertial forward-backward splitting method for solving inclusion problems in Hilbert spaces*, J. Fixed Point Theory Appl., **20** (2018), 17 pages. 1
[9] W. Cholamjiak, S. Das, *A modified projective forward-backward splitting algorithm for variational inclusion problems to predict Parkinson's disease*, Appl. Math. Sci. Eng., **32** (2024), 16 pages. 1
[10] P. Cholamjiak, Y. Shehu, *Inertial forward-backward splitting method in Banach spaces with application to compressed sensing*, Appl. Math., **64** (2019), 409–435. 1
[11] W. Cholamjiak, Y. Shehu, J.-C. Yao, *Prediction of breast cancer through fast optimization techniques applied to machine learning*, Optimization, 1–29. 1
[12] P. L. Combettes, V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., **4** (2005), 1168–1200. 1
[13] Q. Dong, D. Jiang, P. Cholamjiak, Y. Shehu, *A strong convergence result involving an inertial forward–backward algorithm for monotone inclusions*, J. Fixed Point Theory Appl., **19** (2017), 3097–3118. 1
[14] A. Hanjing, S. Suantai, *A fast image restoration algorithm based on a fixed point and optimization method*, Mathematics, **8** (2020), 13 pages 1, 2
[15] S. Ishikawa, *Fixed points by a new iteration method*, Proc. Amer. Math. Soc., **44** (1974), 147–150.
[16] W. M. Jinjri, P. Keikhosrokiani, N. L. Abdullah, *Machine learning algorithms for the classification of cardiovascular disease-A comparative study*, In: 2021 International Conference on Information Technology (ICIT), IEEE, 132–138. 1
[17] J. S. Jung, *Strong convergence of a path for convex minimization, generalized split feasibility, and fixed point problems*, Appl. Set-Valued Anal. Optim., **5** (2023). 1
[18] K. Kankam, P. Cholamjiak, *Strong convergence of the forward-backward splitting algorithms via linesearches in Hilbert spaces*, Appl. Anal. **102** (2023), 1394–1413. 1, 1
[19] K. Kankam, N. Pholasa, P. Cholamjiak, *Hybrid forward-backward algorithms using linesearch rule for minimization problem*, Thai J. Math., **17** (2019), 607–625. 1
[20] K. Kankam, N. Pholasa, P. Cholamjiak, *On convergence and complexity of the modified forward-backward method involving new linesearches for convex minimization*, Math. Methods Appl. Sci., **42** (2019), 1352–1362. 1
[21] S. Kesornprom, K. Kankam, P. Inkrong, N. Pholasa, P. Cholamjiak, *A variant of the proximal gradient method for constrained convex minimization problems*, J. Nonlinear Funct. Anal., **2024** (2024), 16 pages. 1
[22] S. A. Khan, S. Suantai, W. Cholamjiak, *Shrinking projection methods involving inertial forward-backward splitting methods for inclusion problems*, Rev. R. Acad. Cienc. Exactas Fís. Nat. Ser. A Mat. RACSAM, **113** (2019), 645–656. 1
[23] A. Moudafi, M. Oliny, *Convergence of a splitting inertial proximal method for monotone operators*, J. Comput. Appl. Math., **155** (2003), 447–454.
[24] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk SSSR, **269** (1983), 543–547. 1
[25] P. Phairatchatniyom, H. ur. Rehman, J. Abubakar, P. Kumam, J. Martinez-Moreno, *An inertial iterative scheme for solving split variational inclusion problems in real Hilbert spaces*, Bangmod Int. J. Math. Comput. Sci., **7** (2021), 35–52. 1
[26] B. T. Polyak, *Some methods of speeding up the convergence of iteration methods*, USSR Comput. Math. Math. Phys., **4** (1964), 1–17. 1
[27] S. Suantai, P. Jailoka, A. Hanjing, A. (2021). *An accelerated viscosity forward-backward splitting algorithm with the linesearch process for convex minimization problems*, J. Inequal. Appl., **2021** (2021), 19 pages. 1
[28] S. Suantai, K. Kankam, P. Cholamjiak, *A novel forward-backward algorithm for solving convex minimization problem in Hilbert spaces*, Mathematics, **8** (2020), 13 pages. 1, 1
[29] S. Suantai, K. Kankam, P. Cholamjiak, *A projected forward-backward algorithm for constrained minimization with applications to image inpainting*, Mathematics, **9** (2021), 14 pages. 1
[30] S. Suantai, M. A. Noor, K. Kankam, P. Cholamjiak, *Novel forward-backward algorithms for optimization and applications to compressive sensing and image inpainting*, Adv. Differ. Equ., **2021** (2021), 22 pages.
[31] P. Thongpaen, W. Inthakon, A. Kaewkhao, S. Suantai, *Convex minimization problems based on an accelerated fixed point algorithm with applications to image restoration problems*, J. Nonlinear Var. Anal., **7** (2023), 87–101. 1
[32] P. Tseng, *A modified forward-backward splitting method for maximal monotone mappings*, SIAM J. Control Optim., **38** (2000), 431–446. 1
[33] G. M. Vijayalakshmi, M. Ariyanatchi, *Adams-Bashforth Moulton numerical approach on dengue fractional Atangana Baleanu Caputo model and stability analysis*, Int. J. Appl. Comput. Math., **10** (2024), 13 pages. 1
[34] G. M. Vijayalakshmi, P. Roselyn Besi, *ABC fractional order vaccination model for Covid-19 with self-protective measures*, Int. J. Appl. Comput. Math., **8** (2022), 25 pages.

[35] G. M. Vijayalakshmi, P. Roselyn Besi, *Vaccination control measures of an epidemic model with long-term memristive effect*, J. Comput. Appl. Math., **419** (2023), 14 pages.

[36] G. M. Vijayalakshmi, P. Roselyn Besi, A. Kalaivani, G. Sujitha, S. Mahesh, *Microbial coinfections in COVID-19: mathematical analysis using Atangana–Baleanu–Caputo type*, Multiscale Multidiscip. Model. Exp. Des., **7** (2014), 4097–4116. 1