Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(2): 2541–2561.
DOI: 10.3934/math.2025118
Received: 20 November 2024
Revised: 10 January 2025
Accepted: 15 January 2025
Published: 12 February 2025

Research article

An algorithm for variational inclusion problems including quasi-nonexpansive mappings with applications in osteoporosis prediction

Raweerote Suparatulatorn[1,2], Wongthawat Liawrungrueang[3], Thanasak Mouktonglang[1] and Watcharaporn Cholamjiak[4,*]

[1] Department of Mathematics, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand

[2] Office of Research Administration, Chiang Mai University, Chiang Mai 50200, Thailand

[3] Department of Orthopaedics, School of Medicine, University of Phayao, Phayao 56000, Thailand

[4] School of Science, University of Phayao, Phayao 56000, Thailand

* **Correspondence:** Email: watchataporn.ch@up.ac.th.

Abstract: This paper has proposed a novel algorithm for solving fixed point problems for quasi-nonexpansive mappings and variational inclusion problems within a real Hilbert space. The proposed method exhibits weak convergence under reasonable assumptions. Furthermore, we applied this algorithm for data classification to osteoporosis risk prediction, utilizing an extreme learning machine. From the experimental results, our proposed algorithm consistently outperforms existing algorithms across multiple evaluation metrics. Specifically, it achieved higher accuracy, precision, and F1-score across most of the training boxes compared to other methods. The area under the curve (AUC) values from the receiver operating characteristic (ROC) curves further validated the effectiveness of our approach, indicating superior generalization and classification performance. These results highlight the efficiency and robustness of our proposed algorithm, demonstrating its potential for enhancing osteoporosis risk-prediction models through improved convergence and classification capabilities.

Keywords: variational inclusion problem; fixed point problem; osteoporosis; process innovation
Mathematics Subject Classification: 47J25, 49M37, 90C90

1. Introduction

In this paper, let $\mathcal{H}$ represent a real Hilbert space equipped with the inner product $\langle \cdot, \cdot \rangle$ and its corresponding norm $\|\cdot\|$. We denote the sets of real numbers and positive integers by $\mathbb{R}$ and $\mathbb{N}$, respectively.

A fixed point problem involves finding

$$v \in \mathcal{H} \text{ such that } Uv = v, \tag{1.1}$$

where U is a self-mapping on $\mathcal{H}$. We denote the set of solutions to the fixed point problem of the mapping U by $Fix(U)$. Iterative methods are crucial in fixed point theory, especially when dealing with nonlinear mappings. The Picard iteration method [1], developed by Charles Émile Picard in the late 19th century, has long been a cornerstone of mathematical analysis, particularly in solving differential equations. It is highly effective for contractive mappings, where it guarantees convergence to a unique solution by repeatedly applying the function to an initial guess. However, its effectiveness is limited when dealing with nonexpansive mappings, where convergence is not always assured. To address this limitation, Mann [2] introduced the Mann iteration method in 1953, broadening the scope of the Picard method to include nonexpansive mappings. Since then, the Mann iteration has become an essential tool in fixed point theory and numerical analysis, particularly in cases where traditional methods might struggle to converge. In 2013, Khan [3] introduced the Picard-Mann iteration method, a hybrid approach that combines elements of both the Picard and Mann iterations. This method was designed to enhance the convergence of traditional techniques, particularly for certain types of mappings where standard methods can be slow or less effective. By blending the simplicity of Picard's method with the flexibility of Mann's, the Picard-Mann iteration offers a more efficient way to find fixed points, especially in more complex or nonexpansive scenarios. This hybrid approach not only expands its applicability but also has the potential to accelerate convergence compared to using either method alone. Khan further demonstrated both weak and strong convergence results for nonexpansive mappings in uniformly convex Banach spaces using the Picard-Mann iteration method.

A variational inclusion problem (VIP) is typically formulated as follows:

$$\text{Find } v \in \mathcal{H} \text{ such that } 0 \in (X + Y)v, \tag{1.2}$$

where $X : \mathcal{H} \to \mathcal{H}$ is a single-valued mapping and $Y : \mathcal{H} \to 2^{\mathcal{H}}$ is a multivalued mapping. The set of all solutions to the VIP is denoted by $(X + Y)^{-1}(0)$. The study of solving the VIP has garnered significant attention among researchers in optimization and functional analysis. Numerous researchers have focused on developing efficient algorithms for addressing this problem. Two methods that have gained considerable popularity are the forward-backward splitting method [4,5] and the Tseng splitting method [6]. Various researchers have modified and adapted the forward-backward splitting method to solve the VIP (see [7–9]). The Tseng splitting method is another approach that has been refined and applied to solve the VIP (see [10–12]). Both methodologies are extensively referenced in the academic literature on solving the VIP, owing to their efficiency and flexibility in application to various problem formulations. Researchers have further developed these methods, enhancing convergence rates, computational error tolerance, and the capacity to accommodate additional problem constraints. In 2018, Gibali and Thong [13] developed a modified iterative method aimed at solving the VIP. They enhanced existing techniques by adjusting the step size rule based on Tseng's method and the Mann iteration method, to improve both efficiency and applicability. The method is detailed as follows:

Algorithm 1 : Mann Tseng-type method

Initialization: Let $a_1 \in \mathcal{H}$, $\mu_1 > 0$, and $\mu, \alpha_n, \beta_n \in (0, 1)$.

Iterative Steps: Given $n \geq 1$, calculate a_{n+1} as follows:

Step 1. Compute

$$y_n = (I + \mu_n Y)^{-1}(I - \mu_n X)a_n.$$

If $a_n = y_n$, then stop and y_n is a solution of the VIP. Otherwise,

Step 2. Compute

$$z_n = y_n - \mu_n(Xy_n - Xa_n)$$

and

$$a_{n+1} = (1 - \alpha_n - \beta_n)a_n + \beta_n z_n.$$

Step 3. Update

$$\mu_{n+1} = \begin{cases} \min\left\{\mu_n, \dfrac{\mu\|a_n - y_n\|}{\|Xa_n - Xy_n\|}\right\} & \text{if } Xa_n - Xy_n \neq 0; \\ \mu_n & \text{otherwise.} \end{cases}$$

Set $n = n + 1$ and go to **Step 1.**

Under the right conditions, this approach demonstrates strong convergence and provides practical benefits, making it particularly valuable in real-world scenarios. Prior to this, in 1964, Polyak [14] proposed the inertial extrapolation technique, known as the heavy ball method, to accelerate the convergence of iterative algorithms. In 2020, Padcharoen et al. [11] introduced the following splitting method that builds on Tseng's approach and incorporates the inertial extrapolation technique for solving the VIP.

Algorithm 2 : Inertial Tseng method

Initialization: Let $a_0, a_1 \in \mathcal{H}$, $\theta_n \in [0, 1)$, and $\mu_n \in (0, 1/L)$ where L is a Lipschitz constant.
Iterative Steps: Given $n \geq 1$, calculate a_{n+1} as follows:
Step 1. Compute

$$w_n = a_n + \theta_n(a_n - a_{n-1})$$

and

$$y_n = (I + \mu_n Y)^{-1}(I - \mu_n X)w_n.$$

If $w_n = y_n$, then stop and y_n is a solution of the VIP. Otherwise,
Step 2. Compute

$$a_{n+1} = y_n - \mu_n(Xy_n - Xw_n).$$

Set $n = n + 1$ and go to **Step 1.**

While weak convergence was confirmed under typical conditions, the method has also proven effective in practical applications, such as image deblurring and recovery.

In this study, we are interested in investigating the fixed point problem and the VIP, that is,

$$v \in Fix(U) \cap (X + Y)^{-1}(0), \tag{1.3}$$

where $Y : \mathcal{H} \to 2^{\mathcal{H}}$ is a maximal monotone mapping, $X : \mathcal{H} \to \mathcal{H}$ is ℓ-Lipschitz continuous and a monotone mapping, and $U : \mathcal{H} \to \mathcal{H}$ is quasi-nonexpansive and a demiclosed mapping. We denote the set of solutions to this problem by Ψ. Recently, Mouktonglang et al. [15] introduced the following method for solving fixed points of demicontractive mapping and the VIP in the case where $X = \nabla f$ and $Y = \partial g$, where $f : \mathcal{H} \to \mathbb{R}$ and $g : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ are two proper, lower semicontinuous, and convex functions.

Algorithm 3 : Double inertial proximal gradient Mann method

Initialization: Select arbitrary elements $a_0, a_1 \in \mathcal{H}$. Let $\mu \in (0,1)$, $\mu_1 \in (0,\infty)$, $\{\alpha_n\} \subset (0,1)$, $\{\theta_n\} \subset [0,\infty)$, $\{\delta_n\} \subset [0,\infty)$, $\{p_n\} \subset [0,\infty)$, and $\{q_n\} \subset [1,\infty)$.

Iterative Steps: Construct $\{a_n\}$ by using the following steps:

Step 1. Compute

$$\begin{aligned} z_n &= a_n + \theta_n(a_n - a_{n-1}),\\ w_n &= z_n + \delta_n(z_n - a_{n-1}),\\ y_n &= \operatorname{prox}_{\mu_n g}(I - \mu_n \nabla f)w_n,\\ u_n &= y_n + \mu_n(\nabla f(w_n) - \nabla f(y_n)), \end{aligned}$$

and

$$a_{n+1} = (1-\alpha_n)u_n + \alpha_n U u_n.$$

If $w_n = y_n = u_n = Uu_n$, then stop and w_n is a solution of the problem. Otherwise,

Step 2. Update

$$\mu_{n+1} = \begin{cases} \min\left\{\dfrac{\mu q_n \|w_n - y_n\|}{\|\nabla f(w_n) - \nabla f(y_n)\|}, \mu_n + p_n\right\} & \text{if } \nabla f(w_n) \neq \nabla f(y_n);\\ \mu_n + p_n & \text{otherwise.} \end{cases}$$

Replace n with $n+1$ and then repeat **Step 1.**

The authors proved a weak convergence theorem under specific conditions using this method and demonstrated it with a numerical example in signal recovery. The approach is inspired by the proximal gradient technique, double inertial steps, and Mann iteration.

In this article, we present a novel algorithm that demonstrates weak convergence to a common solution for fixed point problems involving quasi-nonexpansive mappings and variational inclusion problems within the context of real Hilbert spaces, under reasonable assumptions. The presented algorithm is given along with essential assumptions in Section 3. Additionally, in Section 4, we employ this algorithm in conjunction with an extreme learning machine for data classification, specifically to predict osteoporosis risk.

2. Preliminaries

We gather essential definitions and lemmas needed to establish our main results. We denote weak and strong convergence as $\rightharpoonup$ and $\to$, respectively. Assume $a, b \in \mathcal{H}$, and we have

$$\|a+b\|^2 = \|a\|^2 + 2\langle a,b\rangle + \|b\|^2, \tag{2.1}$$

$$\|\gamma a + (1-\gamma)b\|^2 = \gamma\|a\|^2 + (1-\gamma)\|b\|^2 - \gamma(1-\gamma)\|a-b\|^2, \tag{2.2}$$

for any $\gamma \in \mathbb{R}$.

Definition 2.1. A self-mapping $X : \mathcal{H} \to \mathcal{H}$ is called

(i) ℓ-Lipschitz continuous if there is $\ell > 0$ such that $\|Xa - Xb\| \leq \ell\|a - b\|$ for all $a, b \in \mathcal{H}$;

(ii) nonexpansive if X is 1-Lipschitz continuous;

(iii) quasi-nonexpansive if $Fix(X)$ is nonempty and $\|Xa - r\| \leq \|a - r\|$ for all $a \in \mathcal{H}$ and $r \in Fix(X)$;

(iv) demiclosed if for any sequence $\{a_n\} \subset \mathcal{H}$, the following implication holds:

$$a_n \rightharpoonup r \text{ and } (I - X)a_n \to 0 \implies r \in Fix(X).$$

Definition 2.2. Let $Y : \mathcal{H} \to 2^{\mathcal{H}}$ be a multivalued mapping. Then Y is said to be

(i) monotone if for all $(a, c), (b, d) \in graph(Y)$ (the graph of mapping Y), $\langle c - d, a - b\rangle \geq 0$;

(ii) maximal monotone if for every $(a, c) \in \mathcal{H} \times \mathcal{H}$, $\langle c - d, a - b\rangle \geq 0$ for all $(b, d) \in graph(Y)$ if and only if $(a, c) \in graph(Y)$.

Lemma 2.3. [16] *Let $X : \mathcal{H} \to \mathcal{H}$ be a mapping and $Y : \mathcal{H} \to 2^{\mathcal{H}}$ a maximal monotone mapping. If $T_\mu := (I + \mu Y)^{-1}(I - \mu X)$ with $\mu > 0$, then $Fix(T_\mu) = (X + Y)^{-1}(0)$.*

Lemma 2.4. [17] *If $Y : \mathcal{H} \to 2^{\mathcal{H}}$ is a maximal monotone mapping and $X : \mathcal{H} \to \mathcal{H}$ is a Lipschitz continuous monotone mapping, then the sum $X + Y$ is also maximal monotone.*

3. Main results

To analyze the convergence, we assume the following conditions.

($C1$) $Y : \mathcal{H} \to 2^{\mathcal{H}}$ is a maximal monotone mapping.

($C2$) $X : \mathcal{H} \to \mathcal{H}$ is ℓ-Lipschitz continuous and a monotone mapping.

($C3$) $U : \mathcal{H} \to \mathcal{H}$ is quasi-nonexpansive and a demiclosed mapping.

($C4$) Ψ is nonempty.

The following algorithm will be employed for Theorem 3.2.

Algorithm 4

Initialization: Select arbitrary elements $a_{-1}, a_0, a_1 \in \mathcal{H}$. Let $\{\alpha_n\} \subset (0, 1)$, $\{\mu_n\} \subset (0, \frac{1}{\ell})$, $\{\theta_n\}, \{\delta_n\} \subset (-\infty, \infty)$, and set $n := 1$.

Iterative Steps: Construct $\{a_n\}$ by using the following steps:

Step 1. Define

$$d_n = a_n + \theta_n(a_n - a_{n-1}) + \delta_n(a_{n-1} - a_{n-2}).$$

Step 2. Compute

$$c_n = (I + \mu_n Y)^{-1}(I - \mu_n X)d_n,$$
$$b_n = c_n + \mu_n(Xd_n - Xc_n).$$

Step 3. Evaluate

$$a_{n+1} = U[\alpha_n d_n + (1 - \alpha_n)Ub_n].$$

If $Ub_n = b_n = c_n = d_n$, then stop and $d_n \in \Psi$. Otherwise, replace n by $n + 1$ and then repeat **Step 1.**

Remark 3.1. Suppose that U and X are mappings on $\mathcal{H}$ and the item $(C1)$ holds. According to Lemma 2.3, if $Ub_n = b_n = c_n = d_n$ in Algorithm 4, then it is easy to show that $d_n \in \Psi$.

Next, we are ready to prove Theorem 3.2.

Theorem 3.2. *Let the sequence $\{a_n\}$ be generated by Algorithm 4 satisfying the items $(C1)$–$(C4)$. Assume that the following conditions are satisfied:*

$(C5)$ $0 < \liminf\limits_{n\to\infty} \alpha_n \le \limsup\limits_{n\to\infty} \alpha_n < 1$.

$(C6)$ $0 < \liminf\limits_{n\to\infty} \mu_n \le \limsup\limits_{n\to\infty} \mu_n < \dfrac{1}{\ell}$.

$(C7)$ $\sum\limits_{n=1}^{\infty} |\theta_n| \|a_n - a_{n-1}\| < \infty$.

$(C8)$ $\sum\limits_{n=1}^{\infty} |\delta_n| \|a_{n-1} - a_{n-2}\| < \infty$.

Then, $\{a_n\}$ converges weakly to a solution of Ψ.

Proof. Let $\tilde{a} \in \Psi$. By the condition $(C6)$, there are $n_0 \in \mathbb{N}, \mu > 0$, and $\bar{\mu} < \frac{1}{\ell}$ such that $\mu \le \mu_n \le \bar{\mu}$ for all $n \ge n_0$. We will now establish the following claims.

Claim 1. For any $n \in \mathbb{N}$,

$$\langle c_n - \tilde{a}, b_n - d_n \rangle \le 0.$$

By using the definition of c_n, we have

$$(I - \mu_n X) d_n \in (I + \mu_n Y) c_n.$$

Thus, we can write

$$y_n = \frac{1}{\mu_n} \left(d_n - c_n - \mu_n X d_n \right),$$

where $y_n \in Y c_n$. Since $X + Y$ is maximal monotone, we obtain

$$\langle c_n - \tilde{a}, X c_n + y_n \rangle \ge 0,$$

implying that

$$\langle c_n - \tilde{a}, c_n + \mu_n (X d_n - X c_n) - d_n \rangle \le 0.$$

Claim 2. For each $n \ge n_0$,

$$\|U b_n - \tilde{a}\|^2 \le \|d_n - \tilde{a}\|^2 - [1 - (\bar{\mu}\ell)^2] \|d_n - c_n\|^2.$$

From (2.1) and the fact that U is a quasi-nonexpansive mapping, we have

$$\begin{aligned} \|U b_n - \tilde{a}\|^2 &\le \|b_n - \tilde{a}\|^2 \\ &= \|c_n - \tilde{a}\|^2 + 2\mu_n \langle c_n - \tilde{a}, X d_n - X c_n \rangle + \mu_n^2 \|X d_n - X c_n\|^2 \end{aligned}$$

$$\begin{aligned}
&= \|c_n - d_n\|^2 + 2\langle c_n - d_n, d_n - \tilde{a}\rangle + \|d_n - \tilde{a}\|^2 + 2\mu_n\langle c_n - \tilde{a}, Xd_n - Xc_n\rangle \\
&\quad + \mu_n^2\|Xd_n - Xc_n\|^2 \\
&= \|d_n - \tilde{a}\|^2 - \|c_n - d_n\|^2 + 2\langle c_n - \tilde{a}, c_n - d_n\rangle + 2\mu_n\langle c_n - \tilde{a}, Xd_n - Xc_n\rangle \\
&\quad + \mu_n^2\|Xd_n - Xc_n\|^2 \\
&= \|d_n - \tilde{a}\|^2 - \|c_n - d_n\|^2 + \mu_n^2\|Xd_n - Xc_n\|^2 + 2\langle c_n - \tilde{a}, b_n - d_n\rangle.
\end{aligned}$$

Using Claim 1, we get

$$\|Ub_n - \tilde{a}\|^2 \leq \|d_n - \tilde{a}\|^2 - \|c_n - d_n\|^2 + \mu_n^2\|Xd_n - Xc_n\|^2.$$

By the Lipschitz continuity of X, we have that

$$\|Ub_n - \tilde{a}\|^2 \leq \|d_n - \tilde{a}\|^2 - \|c_n - d_n\|^2 + (\bar{\mu}\ell)^2\|d_n - c_n\|^2.$$

Thus, Claim 2 is established.

Claim 3. $\lim_{n\to\infty}\|a_n - \tilde{a}\| = \lim_{n\to\infty}\|d_n - \tilde{a}\| = \lim_{n\to\infty}\|e_n - \tilde{a}\|$, where $e_n = \alpha_n d_n + (1-\alpha_n)Ub_n$.
Since U is a quasi-nonexpansive mapping and using Claim 2, we have

$$\begin{aligned}
\|a_{n+1} - \tilde{a}\| &= \|Ue_n - \tilde{a}\| \\
&\leq \|e_n - \tilde{a}\| \\
&\leq \alpha_n\|d_n - \tilde{a}\| + (1-\alpha_n)\|Ub_n - \tilde{a}\| \\
&\leq \|d_n - \tilde{a}\| \text{ for all } n \geq n_0 \\
&\leq \|a_n - \tilde{a}\| + |\theta_n|\|a_n - a_{n-1}\| + |\delta_n|\|a_{n-1} - a_{n-2}\| \text{ for all } n \geq n_0.
\end{aligned}$$

Applying this to Lemma 1 in [18] with the conditions ($C7$) and ($C8$), we derive that the sequence $\{\|a_n - \tilde{a}\|\}$ converges and hence $\lim_{n\to\infty}\|a_n - \tilde{a}\| = \lim_{n\to\infty}\|d_n - \tilde{a}\| = \lim_{n\to\infty}\|e_n - \tilde{a}\|$. In particular, $\{a_n\}$, $\{d_n\}$, and $\{e_n\}$ are bounded.

Claim 4. $\lim_{n\to\infty}\|d_n - Ub_n\| = 0$.
From (2.2), U is a quasi-nonexpansive mapping, and using Claim 2, we have

$$\begin{aligned}
\|a_{n+1} - \tilde{a}\|^2 &\leq \|e_n - \tilde{a}\|^2 \\
&= \alpha_n\|d_n - \tilde{a}\|^2 + (1-\alpha_n)\|Ub_n - \tilde{a}\|^2 - \alpha_n(1-\alpha_n)\|d_n - Ub_n\|^2 \\
&\leq \|d_n - \tilde{a}\|^2 - \alpha_n(1-\alpha_n)\|d_n - Ub_n\|^2 \text{ for all } n \geq n_0.
\end{aligned}$$

This together with Claim 3 and the condition ($C5$) implies that $\lim_{n\to\infty}\|d_n - Ub_n\| = 0$.

Claim 5. $\lim_{n\to\infty}\|Ub_n - b_n\| = 0$.
Again, using Claim 2, we get, for all $n \geq n_0$,

$$[1 - (\bar{\mu}\ell)^2]\|d_n - c_n\|^2 \leq \|d_n - \tilde{a}\|^2 - \|Ub_n - \tilde{a}\|^2.$$

Thus, we obtain from Claim 4 and $1 - (\bar{\mu}\ell)^2 > 0$ that

$$\lim_{n\to\infty}\|d_n - c_n\| = 0. \tag{3.1}$$

By the Lipschitz continuity of X, it follows that

$$\begin{aligned}\|d_n - b_n\| &\leq \|d_n - c_n\| + \|c_n - b_n\| \\ &= \|d_n - c_n\| + \mu_n\|Xd_n - Xc_n\| \\ &\leq (1 + \bar{\mu}\ell)\|d_n - c_n\| \text{ for all } n \geq n_0,\end{aligned}$$

which by (3.1) yields

$$\lim_{n\to\infty} \|d_n - b_n\| = 0. \tag{3.2}$$

Combining (3.2) and Claim 4, we have that

$$\|Ub_n - b_n\| \leq \|Ub_n - d_n\| + \|d_n - b_n\| \to 0 \text{ as } n \to \infty.$$

Claim 6. $\lim_{n\to\infty} \|a_n - b_n\| = \lim_{n\to\infty} \|a_n - c_n\| = 0.$
By using the the definition of d_n, the following inequalities are obtained:

$$\begin{aligned}\|a_n - b_n\| &\leq \|a_n - d_n\| + \|d_n - b_n\| \\ &\leq |\theta_n|\|a_n - a_{n-1}\| + |\delta_n|\|a_{n-1} - a_{n-2}\| + \|d_n - b_n\|\end{aligned}$$

and

$$\begin{aligned}\|a_n - c_n\| &\leq \|a_n - d_n\| + \|d_n - c_n\| \\ &\leq |\theta_n|\|a_n - a_{n-1}\| + |\delta_n|\|a_{n-1} - a_{n-2}\| + \|d_n - c_n\|.\end{aligned}$$

Combining (3.1)–(3.2) and the conditions $(C7)$–$(C8)$, we deduce that Claim 6 is true.
Claim 7. Every weak sequential cluster point of $\{a_n\}$ belongs to Ψ.
Let a^* be a weak sequential cluster point of $\{a_n\}$. Then $a_{n_k} \rightharpoonup a^*$ as $k \to \infty$ for some subsequence $\{a_{n_k}\}$ of $\{a_n\}$. This implies by Claim 6 that $b_{n_k} \rightharpoonup a^*$ and $c_{n_k} \rightharpoonup a^*$ as $k \to \infty$. It follows, from the fact that U is a demiclosed mapping and Claim 5, that $a^* \in Fix(U)$. Next, we show that $a^* \in (X + Y)^{-1}(0)$. Let $(v, u) \in graph\,(X + Y)$, that is, $u - Xv \in Yv$. It is implied by the definition of c_n that $\frac{1}{\mu_{n_k}}(d_{n_k} - c_{n_k} - \mu_{n_k}Xd_{n_k}) \in Yc_{n_k}$. By the maximal monotonicity of Y, we have

$$\left\langle v - c_{n_k}, u - Xv - \frac{1}{\mu_{n_k}}(d_{n_k} - c_{n_k} - \mu_{n_k}Xd_{n_k})\right\rangle \geq 0.$$

Thus, by the monotonicity of X, we get

$$\begin{aligned}\langle v - c_{n_k}, u\rangle &\geq \left\langle v - c_{n_k}, Xv + \frac{1}{\mu_{n_k}}(d_{n_k} - c_{n_k} - \mu_{n_k}Xd_{n_k})\right\rangle \\ &= \langle v - c_{n_k}, Xv - Xc_{n_k}\rangle + \langle v - c_{n_k}, Xc_{n_k} - Xd_{n_k}\rangle + \frac{1}{\mu_{n_k}}\langle v - c_{n_k}, d_{n_k} - c_{n_k}\rangle \\ &\geq \langle v - c_{n_k}, Xc_{n_k} - Xd_{n_k}\rangle + \frac{1}{\mu_{n_k}}\langle v - c_{n_k}, d_{n_k} - c_{n_k}\rangle.\end{aligned}$$

This result follows from the Lipschitz continuity of X and (3.1), giving us

$$\langle v - a^*, u\rangle = \lim_{k\to\infty} \langle v - c_{n_k}, u\rangle \geq 0,$$

which, combined with the maximal monotonicity of $X + Y$, implies that $a^* \in (X + Y)^{-1}(0)$. Hence, $a^* \in \Psi$. Finally, by Opial's lemma in [16], we conclude that the sequence $\{a_n\}$ converges weakly to a point in Ψ. □

4. Application to osteoporosis risk prediction

Osteoporosis is a major global health issue, particularly affecting the elderly population. It leads to weakened bones and a higher risk of fractures, which can result in significant morbidity, loss of mobility, and increased mortality rates. Machine learning (ML) models can analyze large datasets containing complex variables like demographics, lifestyle, genetics, and bone density, identifying individuals at high risk of osteoporosis earlier. This allows for timely interventions, preventing fractures and other complications.

In this section, we implement our newly proposed algorithm as the optimizer for an extreme learning machine (ELM), originally introduced by Huang et al. [19], to assess osteoporosis risk using a comprehensive dataset from Kaggle*. This dataset provides a detailed overview of health factors that contribute to osteoporosis, including demographic details, lifestyle choices, medical history, and bone health indicators. Its comprehensive nature facilitates the development of machine learning models that can accurately identify individuals at high risk for osteoporosis. By analyzing key factors such as age, gender, hormonal changes, and lifestyle habits, our research significantly advances osteoporosis management and prevention strategies. This predictive capability enables early diagnosis, supporting timely interventions that minimize fracture risk, enhance patient outcomes, and optimize the allocation of healthcare resources. Additionally, the integration of machine learning models with our novel optimizer improves prediction accuracy, representing a significant innovation in the field. Table 1 provides readers with an understanding of the dataset's structure, including a detailed description of its components.

Osteoporosis occurs when bone density significantly decreases, resulting in fragile and brittle bones. It is diagnosed when the T-score is -2.5 or lower, based on a bone density scan (DEXA scan), see Figure 1. At this stage, bones have become considerably more porous and weaker, making them prone to fractures. Osteoporosis typically progresses from osteopenia, an intermediate stage characterized by lower-than-normal bone density but not as severe as osteoporosis. Factors such as age, hormonal changes (e.g., menopause), nutritional deficiencies (e.g., calcium or vitamin D), and lack of physical activity can contribute to the development of osteoporosis.

*https://www.kaggle.com/datasets/amitvkulkarni/lifestylefactors-influencing-osteoporosis

Table 1. The characteristics of the Osteoporosis dataset used for risk prediction, including statistics for each feature such as mean ($\bar{x}$), maximum (Max), minimum (Min), standard deviation (SD), and coefficient of variation (CV).

Feature	Description	$\bar{x}$	Max	Min	SD	CV
Age	The age of the individual in years	39.1011	90	18	21.3554	0.5461
Gender	The individual's gender: Male or Female	1.5066	2	1	0.5001	0.3319
Hormonal Changes	Indicates if the individual has experienced hormonal changes, such as menopause	1.4989	2	1	0.5001	0.3336
Family History	Indicates a family history of osteoporosis or fractures: Yes or No	1.5097	2	1	0.5000	0.3312
Race/Ethnicity	The individual's race or ethnicity: e.g., Caucasian, African American, Asian	2.0255	3	1	0.8183	0.4040
Body Weight	The individual's body weight status: Normal or Underweight	1.4754	2	1	0.4995	0.3385
Calcium Intake	The individual's dietary calcium intake: Low or Adequate	1.5127	2	1	0.4996	0.3304
Vitamin D Intake	The individual's vitamin D intake: Insufficient or Sufficient	1.4836	2	1	0.4998	0.3369
Physical Activity	Indicates the individual's physical activity level: Sedentary for low activity or Active	1.4785	2	1	0.4996	0.3379
Smoking	Indicates if the individual is a smoker: Yes or No	1.4984	2	1	0.5001	0.3337
Alcohol Consumption	Indicates the individual's alcohol consumption: None for non-drinkers or Moderate	1.4954	2	1	0.5001	0.3344
Medical Conditions	Any existing medical conditions the individual may have, such as Rheumatoid Arthritis	2.0158	3	1	0.8226	0.4081
Medications	Any medications the individual is currently taking, such as Corticosteroids or No Medications	1.4969	2	1	0.5001	0.3340
Prior Fractures	Indicates if the individual has previously experienced fractures: Yes or No	1.4979	2	1	0.5001	0.3338
Osteoporosis	The target variable indicating	979 instances of the presence of osteoporosis 979 instances of the absence of osteoporosis				

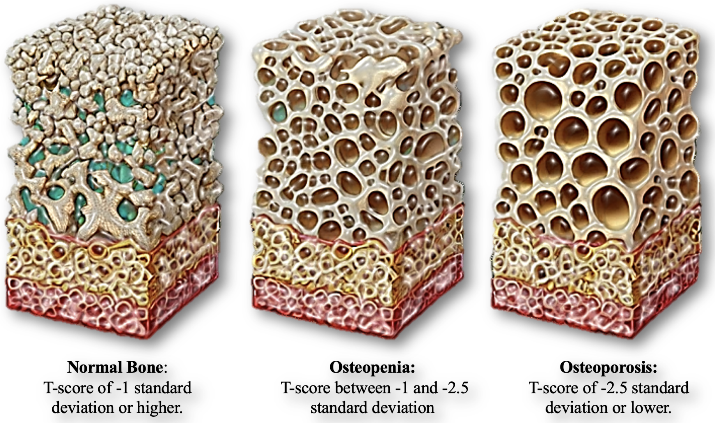
Normal Bone:
T-score of -1 standard deviation or higher.
Osteopenia:
T-score between -1 and -2.5 standard deviation
Osteoporosis:
T-score of -2.5 standard deviation or lower.

Figure 1. The figure illustrates three stages of bone density: (1) Normal bone, which has a T-score of -1 or higher, indicating healthy bone density; (2) Osteopenia, which shows mild bone loss with a T-score between -1 and -2.5, representing a condition of lower-than-normal bone density; and (3) Osteoporosis, characterized by significant bone loss and increased porosity, with a T-score of -2.5 or lower, indicating a higher risk of fractures.

In the process of creating our extreme learning machine (ELM), we consider N distinct samples, where the training set $\mathcal{S} := \{(a_n, t_n) : x_n \in \mathbb{R}^n, t_n \in \mathbb{R}^m, n = 1, 2, \ldots, N\}$ consists of input data x_n and corresponding target outputs t_n. The output function of an ELM for a standard single-layer feedforward network (SLFN) with M hidden nodes is mathematically represented as:

$$O_j = \sum_{i=1}^{M} \beta_i \frac{1}{1 + e^{-(w_i a_j + b_i)}},$$

where w_i is a randomly initialized weight, and b_i is a randomly initialized bias for the i-th hidden node. The goal is to find the optimal output weights β_i. The above system of linear equations can be represented in matrix form as $T = H\beta$, where

$$H = \begin{bmatrix} \frac{1}{1+e^{-(w_1 a_1 + b_1)}} & \cdots & \frac{1}{1+e^{-(w_M a_1 + b_M)}} \\ \vdots & \ddots & \vdots \\ \frac{1}{1+e^{-(w_1 a_N + b_1)}} & \cdots & \frac{1}{1+e^{-(w_M a_N + b_M)}} \end{bmatrix},$$

where H is the hidden layer output matrix, $T = [t_1^T, \ldots, t_N^T]^T$ is the target output matrix, and $\beta = [\beta_1^T, \ldots, \beta_M^T]^T$ is the vector of optimal output weights. These optimal weights can be computed as $\beta = H^\dagger T$, where $H^\dagger$ is the Moore-Penrose generalized inverse of H, though finding $H^\dagger$ may be challenging in practice. Therefore, obtaining a solution β via convex minimization can help address this challenge. The least squares problem is particularly effective for this, and regularization is a commonly employed technique in machine learning and statistics to mitigate overfitting, enhance model generalization, and ultimately improve performance in classification tasks. We conducted a series of experiments on a classification problem, explicitly using the well-known least absolute shrinkage and selection operator (LASSO) method [20]. The detailed descriptions of these experiments are provided below: For $\lambda > 0$,

$$\min_{\beta \in R^M} \frac{1}{2} \|H\beta - T\|_2^2 + \lambda \|\beta\|_1. \tag{4.1}$$

By applying Algorithm 4 to solve the problem (4.1), we are setting $X\beta \equiv \nabla(\frac{1}{2}\|H\beta - T\|_2^2)$ and $Y\beta \equiv \partial(\lambda\|\beta\|_1)$ with $\lambda = 0.01$.

We evaluate the performance of the classification algorithms using four evaluation metrics: Accuracy, precision, recall, and F1-score [21]. These metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%;$$

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%;$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\%;$$

$$\text{F1-score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}.$$

In these formulas, TP represents True Positives, TN True Negatives, FP False Positives, and FN False Negatives.

Additionally, we used binary cross-entropy loss [22] to evaluate the model's ability to distinguish between two classes in binary classification tasks. This loss is computed as the average:

$$\text{Loss} = -\sum_{i=1}^{N} \varphi_i \log \bar{\varphi}_i + (1 - \varphi_i) \log(1 - \bar{\varphi}_i),$$

where $\bar{\varphi}_i$ represents the predicted probability for the i-th instance, φ_i is the corresponding true label, and N is the total number of instances.

Next, as illustrated in Figure 2, we partition the dataset using a 5-fold cross-validation approach. In each fold, 80% of the data is used for training (highlighted in purple), and 20% is allocated for validation (highlighted in green). This ensures that every subset of the data is used for validation exactly once, while the rest is used for training.

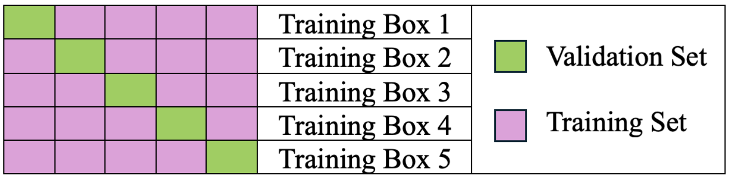

Figure 2. Illustration of 5-fold cross-validation. In each fold, 80% of the data (represented in purple) is used for training, and the remaining 20% (represented in green) is used for validation. Each fold uses a different subset for validation, ensuring that the entire dataset is validated once while the other folds are used for training.

For the comparison of Algorithms 1–3, we set all parameters as follows: The number of hidden nodes is $M = 500$, with randomly initialized weights w_i in the range $[-50, 50]$ and biases b_i in the range $[-5, 5]$. Specifically, we define: $\theta_n = 0.0004$ for Algorithm 2:

$$\theta_n = \begin{cases} \frac{1}{n^3}, & \text{if } a_n \neq a_{n-1} \text{ and } n > N, \\ \theta, & \text{otherwise} \end{cases}$$

with

$$\delta_n = \begin{cases} \frac{1}{n^3}, & \text{if } a_{n-1} \neq a_{n-2} \text{ and } n > N, \\ \delta, & \text{otherwise} \end{cases}$$

for Algorithm 3, and

$$\theta_n = \begin{cases} \frac{1}{\|a_n - a_{n-1}\| n^3}, & \text{if } a_n \neq a_{n-1} \text{ and } n > N, \\ \theta, & \text{otherwise} \end{cases}$$

with

$$\delta_n = \begin{cases} \frac{1}{\|a_{n-1} - a_{n-2}\| n^3}, & \text{if } a_{n-1} \neq a_{n-2} \text{ and } n > N, \\ \delta, & \text{otherwise} \end{cases}$$

for Algorithm 4 where N denotes the iteration number at which we decide to stop the algorithm. For further details on the parameter settings, please refer to Table 2.

Table 2. Parameter settings for Algorithms 1–4. The parameters include μ_1, μ_n, and μ representing step sizes, α_n and β_n for the learning rate, θ and δ for adjustment factors, and p_n and q_n for additional terms related to the iteration process.

	μ_1, μ_n	μ	α_n	β_n	θ	δ	p_n	q_n
Algorithm 1	$\frac{0.999}{\|H\|^2}$	2.5×10^{-6}	$\frac{1}{n+1}$	$\frac{1}{2} - \frac{1}{2n+1}$	-	-	-	-
Algorithm 2	$\frac{0.999}{\|H\|^2}$	-	-	-	-	-	-	-
Algorithm 3	$\frac{0.999}{\|H\|^2}$	0.5	0.9	-	0.999	0.8	$\frac{1}{n^2}$	$1 + \frac{1}{n}$
Algorithm 4	$\frac{0.999}{\|H\|^2}$	-	0.8	-	0.999	-0.001	-	-

The results for each algorithm, evaluated on Training Box 1, are presented in Table 3.

Table 3. Performance comparison of Algorithms 1–4 on the validation set from Training Box 1. These results are based on the model's evaluation using the first fold of the 5-fold cross-validation scheme, where Training Box 1 serves as the validation set.

	Iter.	CPU Time	Accuracy Test	Precision	Recall	F1-score
Algorithm 1	54	1.7352	77.44	76.92	77.72	77.31
Algorithm 2	51	1.6293	77.95	77.94	77.94	77.94
Algorithm 3	45	2.4402	81.79	96.41	74.60	84.11
Algorithm 4	54	2.5853	82.05	91.79	76.82	83.64

Table 4 displays the performance outcomes for each algorithm, assessed using Training Box 2.

Table 4. Performance comparison of Algorithms 1–4 on the validation set from Training Box 2, based on the second fold of the 5-fold cross-validation scheme.

	Iter.	CPU Time	Accuracy Test	Precision	Recall	F1-score
Algorithm 1	42	1.4179	83.89	84.69	83.41	84.05
Algorithm 2	42	1.4632	83.63	84.69	83.00	83.83
Algorithm 3	42	2.2095	85.93	92.85	81.61	86.87
Algorithm 4	35	2.2713	85.93	92.85	81.61	86.87

The performance outcomes for each algorithm, after being tested on Training Box 3, are detailed in Table 5.

Table 5. Performance comparison of Algorithms 1–4 on the validation set from Training Box 3, based on the third fold of the 5-fold cross-validation scheme.

	Iter.	CPU Time	Accuracy Test	Precision	Recall	F1-score
Algorithm 1	36	1.3214	83.89	87.24	81.81	84.44
Algorithm 2	33	1.2631	83.89	87.24	81.81	84.44
Algorithm 3	14	0.6169	84.14	87.24	82.21	84.65
Algorithm 4	28	1.8228	84.40	90.81	80.54	85.37

The performance outcomes for each algorithm, after being tested on Training Box 4, are detailed in Table 6.

Table 6. Performance comparison of Algorithms 1–4 on the validation set from Training Box 4, based on the fourth fold of the 5-fold cross-validation scheme.

	Iter.	CPU Time	Accuracy Test	Precision	Recall	F1-score
Algorithm 1	49	1.6230	76.47	75.51	77.08	76.28
Algorithm 2	46	2.3913	76.98	76.53	77.31	76.92
Algorithm 3	44	2.2669	79.28	83.16	77.25	80.09
Algorithm 4	42	2.6810	82.35	93.36	76.56	84.13

The performance results for each algorithm, evaluated on Training Box 5, are provided in Table 7.

Table 7. Performance comparison of Algorithms 1–4 on the validation set from Training Box 5, based on the fifth fold of the 5-fold cross-validation scheme.

	Iter.	CPU Time	Accuracy Test	Precision	Recall	F1-score
Algorithm 1	51	1.7199	84.30	84.69	83.83	84.26
Algorithm 2	37	1.2250	84.56	85.20	83.91	84.55
Algorithm 3	27	1.4804	85.06	93.36	79.91	86.11
Algorithm 4	42	2.7039	85.32	94.89	79.48	86.51

Remark 4.1. 1) From Tables 3–7, the performance of Algorithms 1–4 was evaluated using a 5-fold cross-validation scheme, with each training box serving as the validation set in turn:

(i) Training Box 1: Algorithm 4 achieved the highest accuracy (82.05%) with notable precision (91.79%) and F1-score (83.64%).

(ii) Training Box 2: Algorithms 3 and 4 tied for the highest accuracy (85.93%), with Algorithm 4 showing marginally better CPU time and recall (81.61%).

(iii) Training Box 3: Algorithm 4 again performed best, achieving an accuracy of 84.40% and a high precision (90.81%).

(iv) Training Box 4: Algorithm 4 had the best accuracy (82.35%) and precision (93.36%).

(v) Training Box 5: Algorithm 4 tied for the highest accuracy (85.32%), and showed the best precision (94.89%) and marginally better number of iterations.

In general, Algorithm 4 consistently demonstrated strong performance in accuracy, precision, and F1-score across most of the training boxes.

2) In practical applications of the proposed algorithm to machine learning problems, the challenge of unknown or difficult-to-estimate Lipschitz constants is mitigated by the finite nature of the feature set. In such cases, the Lipschitz constant can be approximated more quickly due to the boundedness of the feature space. Moreover, the efficiency and convergence of the proposed algorithm are not significantly impacted by the limitations associated with estimating the Lipschitz constant. This is demonstrated through the results presented in Tables 3–7, where the algorithm achieves effective performance and convergence despite potential uncertainties in the Lipschitz parameter.

From Figures 3–7, the accuracy and loss plots for Algorithm 4 across all training boxes show consistent trends, with training and validation accuracy remaining relatively close to each other throughout the iterations, indicating minimal overfitting. The training and validation loss curves also exhibit similar behavior, steadily decreasing over time and stabilizing. However, in some figures (such as Figures 3 and 6), the slight divergence between the training and validation loss can be observed in later iterations, suggesting potential signs of mild overfitting. The model maintains a good balance between training and validation performance, demonstrating that regularization and parameter choices prevent significant overfitting. The performance trends suggest that Algorithm 4 generalizes well to the validation data without overfitting the training set.

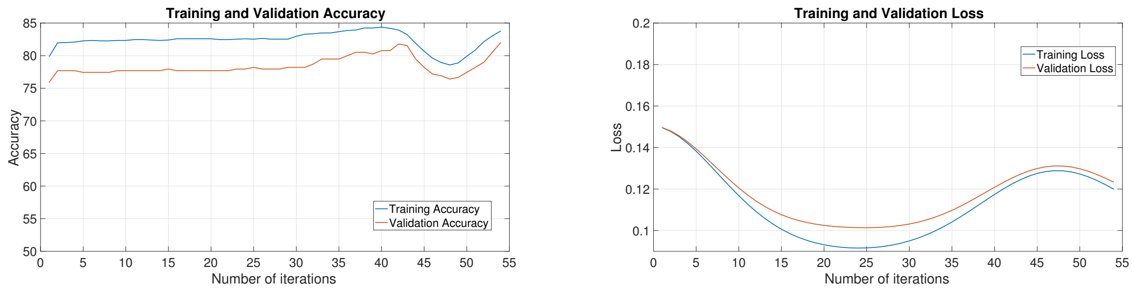

Figure 3. (Left) Training and validation accuracy (left) and loss (right) for Algorithm 4 from Table 3 over 54 iterations. The blue lines represent training performance, while the red lines show validation performance, demonstrating stable accuracy and decreasing loss.

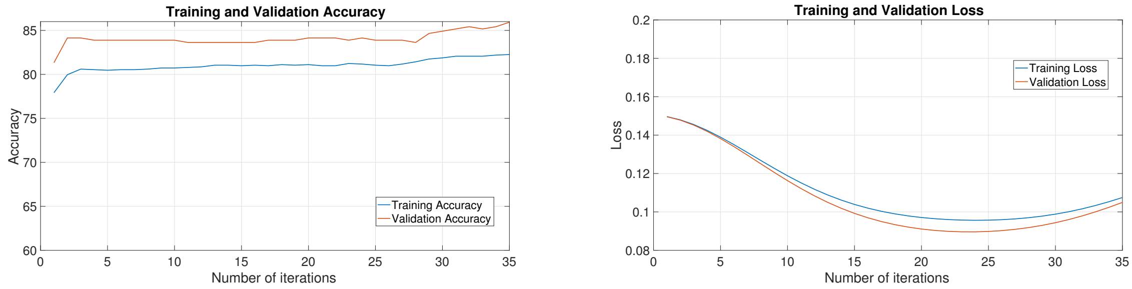

Figure 4. Training and validation accuracy (left) and loss (right) for Algorithm 4 from Table 4 over 35 iterations. The blue lines represent training performance, while the red lines show validation performance, demonstrating stable accuracy and decreasing loss.

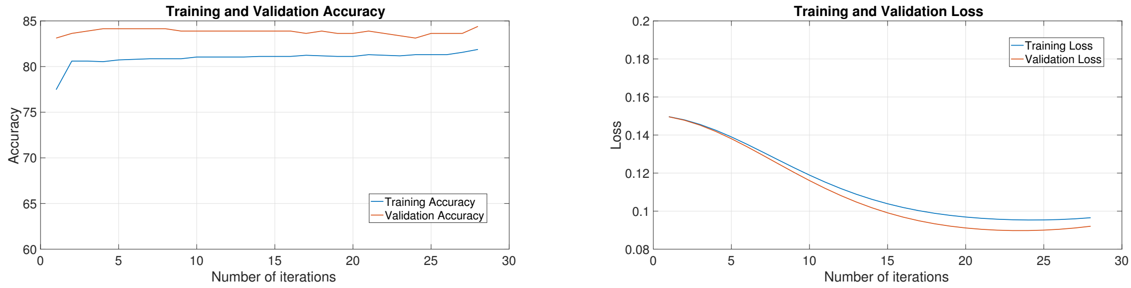

Figure 5. Training and validation accuracy (left) and loss (right) for Algorithm 4 from Table 5 over 28 iterations. The blue lines represent training performance, while the red lines show validation performance, demonstrating stable accuracy and decreasing loss.

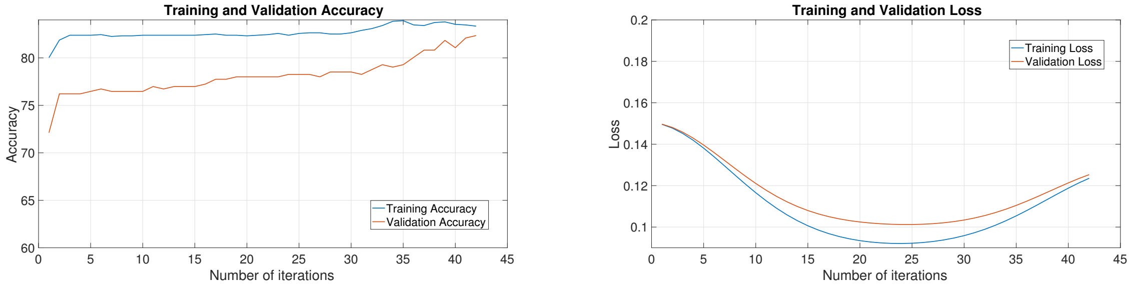

Figure 6. Training and validation accuracy (left) and loss (right) for Algorithm 4 from Table 6 over 42 iterations. The blue lines represent training performance, while the red lines show validation performance, demonstrating stable accuracy and decreasing loss.

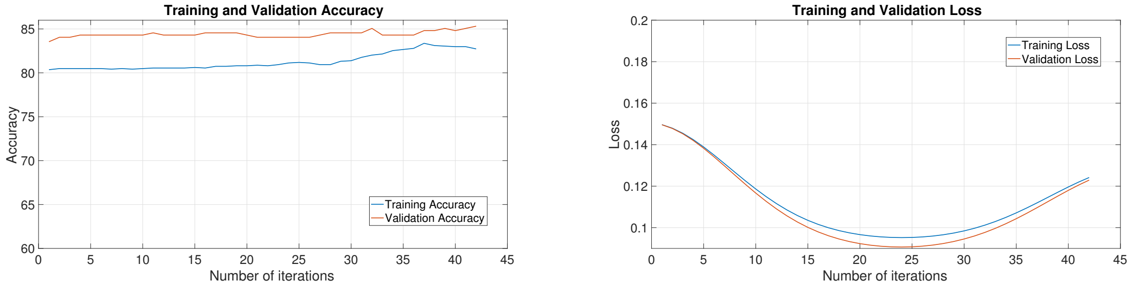

Figure 7. Training and validation accuracy (left) and loss (right) for Algorithm 4 from Table 7 over 42 iterations. The blue lines represent training performance, while the red lines show validation performance, demonstrating stable accuracy and decreasing loss.

From Figure 8, we see that the ROC curves illustrate the model's classification performance, where higher AUC values reflect stronger class separation.

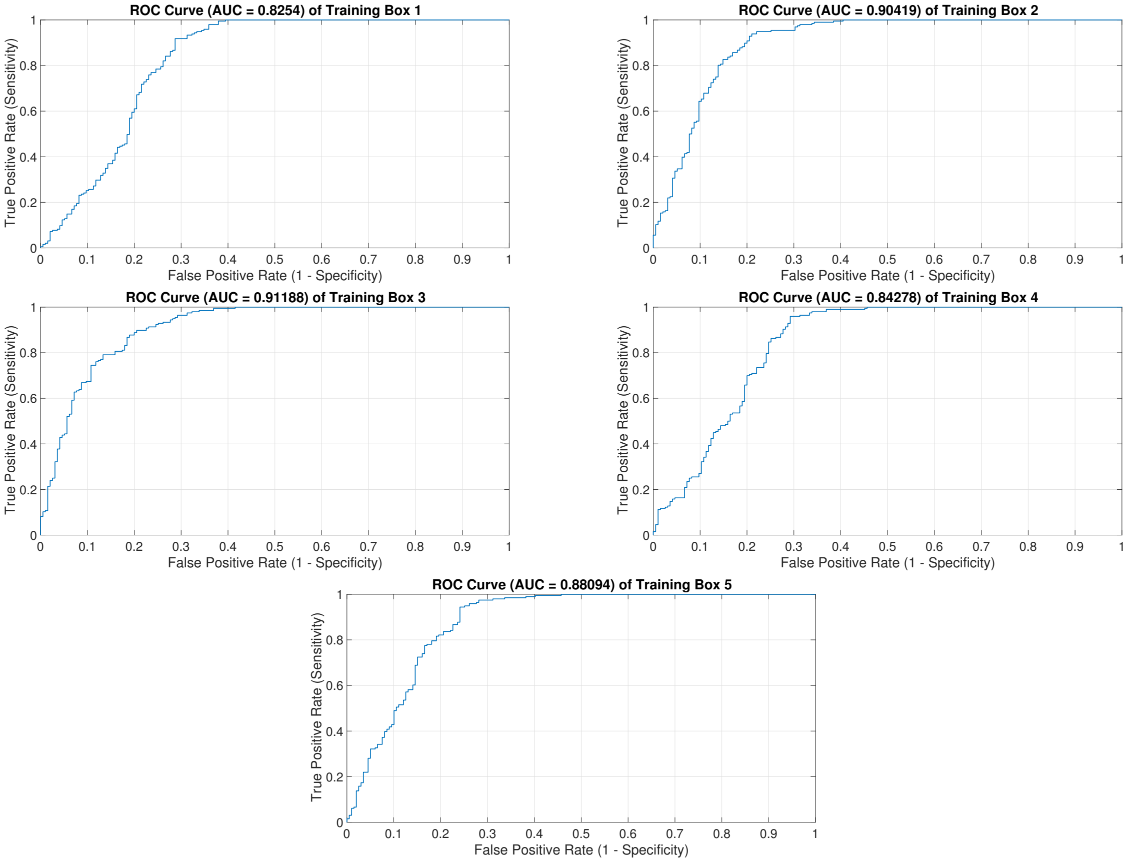

Figure 8. Receiver operating characteristic (ROC) curves for Algorithm 4 across the five training boxes. Each plot represents the ROC curve for the respective training box, with the area under the curve (AUC) values reported as follows: Training Box 1 (AUC = 0.8254), Training Box 2 (AUC = 0.90419), Training Box 3 (AUC = 0.91188), Training Box 4 (AUC = 0.84278), and Training Box 5 (AUC = 0.88094).

5. Conclusions

We proposed a novel algorithm (Algorithm 4) to solve variational inclusion problems and fixed point problems involving quasi-nonexpansive mappings in a real Hilbert space. Our main theorem establishes the weak convergence of the proposed algorithm under certain conditions. We also applied the algorithm in conjunction with an extreme learning machine to the problem of data classification, specifically to predict osteoporosis risk. Our algorithm achieves an accuracy of over 82%, a precision of over 91%, a recall of 76%, and an F1-score of over 83% across all training boxes, which demonstrate the effectiveness of the algorithm we developed.

Data availability

The data are available on the Kaggle website
(https://www.kaggle.com/datasets/amitvkulkarni/lifestylefactors-influencing-osteoporosis).

Institutional Review Board statement

This study was conducted in accordance with the Declaration of Helsinki, the Belmont Report, CIOMS guideline for the International Conference on Harmonization in Good Cinical Pratice, or ICH-GCP or 45CFR 46.101(b), and with approval from the Ethics Committee and Institutional Review Board of the Faculty of the University of Phayao (Institutional Review Board (IRB) approval, IRB Number: HREC-UP-HSST 1.102867).

Author contributions

Raweerote Suparatulatorn: Writing – original draft, formal analysis; Wongthawat Liawrungrueang: Writing – review & editing, data curation; Thanasak Mouktonglang: Writing – review & editing, project administration; Watcharaporn Cholamjiak: Writing – review & editing, software. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research work was partially supported by the CMU Proactive Researcher, Chiang Mai University [grant number 780/2567]. W. Liawrungrueang and W. Cholamjiak would like to thank the Thailand Science Research and Innovation Fund (Fundamental Fund 2025, Grant No. 5025/2567) and the University of Phayao.

Conflict of interest

The authors declare no conflicts of interest.

References

1. E. Picard, Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives, *J. Math. Pure. Appl.,* **6** (1890), 145–210.
2. W. R. Mann, Mean value methods in iteration, *P. Am. Math. Soc.,* **4** (1953), 506–510. https://doi.org/10.2307/2032162
3. S. H. Khan, A Picard-Mann hybrid iterative process, *Fixed Point Theory A.,* **2013** (2013), 69. https://doi.org/10.1186/1687-1812-2013-69
4. P. L. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, *SIAM J. Numer. Anal.,* **16** (1979), 964–979. https://doi.org/10.1137/0716071
5. G. B. Passty, Ergodic convergence to a zero of the sum of monotone operators in Hilbert space, *J. Math. Anal. Appl.,* **72** (1979), 383–390. https://doi.org/10.1016/0022-247X(79)90234-8

6. P. Tseng, A modified forward-backward splitting method for maximal monotone mappings, *SIAM J. Control Optim.,* **38** (2000), 431–446. https://doi.org/10.1137/S0363012998338806

7. D. A. Lorenz, T. Pock, An inertial forward-backward algorithm for monotone inclusions, *J. Math. Imaging Vis.,* **51** (2015), 311–325. https://doi.org/10.1007/s10851-014-0523-2

8. W. Cholamjiak, P. Cholamjiak, S. Suantai, An inertial forward-backward splitting method for solving inclusion problems in Hilbert spaces, *J. Fixed Point Theory A.,* **20** (2018), 42. https://doi.org/10.1007/s11784-018-0526-5

9. P. Peeyada, R. Suparatulatorn, W. Cholamjiak, An inertial Mann forward-backward splitting algorithm of variational inclusion problems and its applications, *Chaos Soliton. Fract.,* **158** (2022), 112048. https://doi.org/10.1016/j.chaos.2022.112048

10. R. Suparatulatorn, A. Khemphet, Tseng type methods for inclusion and fixed point problems with applications, *Mathematics,* **7** (2019), 1175. https://doi.org/10.3390/math7121175

11. A. Padcharoen, D. Kitkuan, W. Kumam, P. Kumam, Tseng methods with inertial for solving inclusion problems and application to image deblurring and image recovery problems, *Comput. Math. Methods,* **3** (2021), e1088. https://doi.org/10.1002/cmm4.1088

12. P. Cholamjiak, D. V. Hieu, Y. J. Cho, Relaxed forward-backward splitting methods for solving variational inclusions and applications, *J. Sci. Comput.,* **88** (2021), 85. https://doi.org/10.1007/s10915-021-01608-7

13. A. Gibali, D. V. Thong, Tseng type methods for solving inclusion problems and its applications, *Calcolo,* **55** (2018), 1–22. https://doi.org/10.1007/s10092-018-0292-1

14. B. Polyak, Some methods of speeding up the convergence of iteration methods, *USSR Comput. Math. Math. Phys.,* **4** (1964), 791–817.

15. T. Mouktonglang, W. Chaiwino, R. Suparatulatorn, A proximal gradient method with double inertial steps for minimization problems involving demicontractive mappings, *J. Inequal. Appl.,* **2024** (2024), 69. https://doi.org/10.1186/s13660-024-02965-y

16. H. H. Bauschke, P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, CMS Books in Mathematics, Springer, New York, 2011.

17. H. Brézis, *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert*, Math. Studies 5, North-Holland, Amsterdam, Netherlands, 1973.

18. A. Auslender, M. Teboulle, S. Ben-Tiba, A logarithmic-quadratic proximal method for variational inequalities, *Comput. Optim. Appl.,* **12** (1999), 31–40.

19. G. B. Huang, Q. Y. Zhu, C. K. Siew, *Extreme learning machine: A new learning scheme of feedforward neural networks*, In: 2004 IEEE International Joint Conference on Neural Networks, 2004, 985–990. https://doi.org/10.1109/IJCNN.2004.1380068

20. R. Tibshirani, Regression shrinkage and selection via the lasso, *J. Roy. Stat. Soc. B,* **58** (1996), 267–288.

21. N. W. S. Wardhani, M. Y. Rochayani, A. Iriany, A. D. Sulistyono, P. Lestantyo, *Cross-validation metrics for evaluating classification performance on imbalanced data*, In: 2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2019, 14–18. https://doi.org/10.1109/IC3INA48034.2019.8949568

22. M. Akil, R. Saouli, R. Kachouri, Fully automatic brain tumor segmentation with deep learning-based selective attention using overlapping patches and multi-class weighted cross-entropy, *Med. Image Anal.,* **63** (2020), 101692. https://doi.org/10.1016/j.media.2020.101692