big data and cognitive computing

MDPI

Article

A Recursive Attribute Reduction Algorithm and Its Application in Predicting the Hot Metal Silicon Content in Blast Furnaces

Zhanqi Li, Pan Cheng, Linzi Yin * and Yuyin Guan

School of Electronic Information, Central South University, Changsha 410083, China; lizhanqi077@gmail.com (Z.L.); i939816238@gmail.com (P.C.); 15717515211@163.com (Y.G.)
* Correspondence: yinlinzi@csu.edu.cn

Abstract: For many complex industrial applications, traditional attribute reduction algorithms are often inefficient in obtaining optimal reducts that align with mechanistic analyses and practical production requirements. To solve this problem, we propose a recursive attribute reduction algorithm that calculates the optimal reduct. First, we present the notion of priority sequence to describe the background meaning of attributes and evaluate the optimal reduct. Next, we define a necessary element set to identify the "individually necessary" characteristics of the attributes. On this basis, a recursive algorithm is proposed to calculate the optimal reduct. Its boundary logic is guided by the conflict between the necessary element set and the core attribute set. The experiments demonstrate the proposed algorithm's uniqueness and its ability to enhance the prediction accuracy of the hot metal silicon content in blast furnaces.

Keywords: attribute reduction; priority sequence; recursive algorithm; silicon content

check for updates

Academic Editor: Salvador García López

Received: 20 November 2024
Revised: 21 December 2024
Accepted: 24 December 2024
Published: 3 January 2025

Citation: Li, Z.; Cheng, P.; Yin, L.; Guan, Y. A Recursive Attribute Reduction Algorithm and Its Application in Predicting the Hot Metal Silicon Content in Blast Furnaces. *Big Data Cogn. Comput.* **2025**, *9*, 6. https://doi.org/10.3390/bdcc9010006

1. Introduction

Introduced by Pawlak in 1982, rough set theory serves as a mathematical tool for dealing with vague, imprecise, and uncertain knowledge, garnering increasing attention in computer sciences, artificial intelligence, medical applications, etc. [1,2]. Attribute reduction, a core aspect of rough set theory, simplifies datasets by eliminating irrelevant attributes, which is commonly known as feature selection in machine learning [3–5]. Over the last decades, researchers have developed many heuristic reduction algorithms based on the positive region [6–9], the discernibility matrix [10,11], and information entropy [12–14]. While these algorithms have achieved efficiency in running time and storage, they are not always effective in obtaining the optimal reduct when used in complex industrial applications, especially in blast furnace smelting, because they overlook the underlying industrial mechanisms embedded in the data. Without these mechanistic insights, current techniques may discard essential attributes or retain unnecessary ones, leading to suboptimal reductions with limited practical utility.

Blast furnace smelting is the most energy-consuming process in iron and steel production [15,16]. Its internal state is difficult to measure directly because of the influence of high temperature and high pressure. Fortunately, extensive research has demonstrated that the silicon content in hot metal exhibits a close relationship with its thermal state [17–19]. Thus, the accurate prediction of the hot metal silicon content is key to the optimal control of blast furnaces.

As a classical complex industrial process, the accuracy of a silicon content prediction depends not only on excellent nonlinear models but also on high-quality datasets. However, due to the strong parameter coupling and nonlinear characteristics of blast furnace systems, original datasets often contain a large number of redundant attributes. It is therefore necessary to calculate a reduction set to improve the training speed and prediction accuracy.

According to rough set theory, there may be multiple complete reduction sets in a dataset. Traditional algorithms, however, only compute one of them at random. From the perspective of blast furnace mechanisms, each attribute has its specific physical meaning and reflects different aspects of the furnace. This means that different reduction sets represent different information, and there should be an optimal reduction set corresponding to both the mechanistic analysis and practical production. Therefore, determining the optimal reduction set represents a meaningful research task.

In addition, how the optimal reduction set of a blast furnace is evaluated is also necessary to consider. In previous studies, many researchers have used the test costs of attributes to evaluate a reduction set, believing that the minimum cost is optimal [20–23]. However, related smelting mechanism analyses and practical production experience have shown that it is hard to set the exact numerical cost for all attributes. Instead, researchers have achieved better results when analyzing which attributes are more important.

Motivated by the above observations, we suggest an importance sequence, called the attribute priority sequence, in this paper to describe the background meaning of the attributes and prior knowledge related to the blast furnace. On this basis, a priority-optimal reduct is defined, and a novel attribute reduction algorithm using recursion technology is proposed to calculate it. Some experimental results on UCI datasets show differences between the proposed algorithm, classical method, and state-of-the-art method. We also applied this algorithm on a real dataset obtained from a blast furnace, and we trained a machine learning model to show the proposed algorithm's performance. The major contributions of this paper are as follows:

(1) We propose a novel heuristic reduction construction.

Existing heuristic reduction algorithms commonly adopt the following three kinds of construction: addition–deletion constructions (Algorithm 1), deletion constructions, and addition constructions. In this paper, we propose a novel recursion construction that is effective in obtaining special reducts, such as the optimal reduct, minimal reduct, and minimal cost reduct.

(2) We define a new optimal reduct.

Traditional rough set theories often treat a minimal reduct as optimal because they ignore related prior knowledge. Many researchers have applied the notion of "cost" to describe prior knowledge and treated a minimal cost reduct as optimal. As mentioned above, "cost" is not always effective in complex situations because it is hard to set the exact costs for all attributes. Therefore, we propose the notion of an attribute priority sequence and define a priority-optimal reduct to represent an optimal reduct. The new definition of the optimal reduct is simple and suitable for complex applications.

(3) We provide a recursive reduction algorithm and illustrate its validity based on UCI datasets and a real application on silicon content prediction.

The proposed algorithm is the first recursion-based reduction algorithm, and the detailed reasoning, as well as the experimental results, show its validity. Furthermore, the proposed algorithm heuristically identifies a priority-optimal reduct.

The rest of this study is organized as follows. Section 2 presents some basic knowledge on Pawlak rough set. Section 3 presents the definition of the priority-optimal reduct and proposes an attribute reduction algorithm using recursion technology to obtain the priority-optimal reduct. Section 4 discusses the experiments and results of hot metal silicon content prediction in blast furnaces.

Algorithm 1. Traditional heuristic attribute reduction algorithm.

Input: Information table S and priority sequence $\{c_1, c_2, \ldots, c_n\}$.
Output: Reduct R.
Step 1: Construct the discernibility matrix, M; calculate the core attribute set, $CORE(M)$; and the delete elements discerned by $CORE(M)$.
Step 2: Delete the attributes belonging to $CORE(M)$ from the priority sequence; then, the new priority sequence is $\{c_{i1}, c_{i2}, \ldots, c_{im}\}$.
Step 3: Addition:
$R' = \varnothing, k = 1.$
While R' is not a super reduct of M, **do**
$$R' = R' \cup c_{ik}, k = k + 1$$
End
Step 4: Deletion:
While $k > 0$, **do**
If $R' - \{c_{ik}\}$ is a super reduct **then**
$$R' = R' - \{c_{ik}\}$$
End
$k = k - 1$
End
Step 5: Output $R = R' \cup CORE(M)$

2. Preliminary Knowledge on Pawlak Rough Set

In rough set theory, data are presented in an information table S [7].

$$S = \langle U, At, \{V_a | a \in At\}, \{I_a | a \in At\} \rangle$$

where U is the universe, At is a finite non-empty set of attributes, V_a is a non-empty set of values of attribute a, and $I_a : U \to V_a$ is an information function that maps an object in U to exactly one value in V_a. As a special type, an information table S is also referred to as a decision table if $At = C \cup D$, where $C = \{c_1, c_2, \ldots, c_n\}$ is the condition attribute set and $D = \{d\}$ is the decision attribute set. A decision table is inconsistent if it contains two objects with the same condition values but different decision values.

Definition 1. *Given a subset of attributes $B \subseteq C$, an indiscernibility relationship $IND(B)$ is defined as follows.*

$$IND(B) = \left\{ (x, x') \in U^2 \ | \forall a \in B, I_a(x) = I_a(x') \right\} \tag{1}$$

The equivalence class (or granule) of object x with respect to C is $[x]_C = \{y \in U \,|(x, y) \in IND(C)\}$. The union of all the granules with respect to C is referred to as a partition of the universe, described as $U/C = \{[x]_C \,| x \in U\}$. Granule $[x]_C$ is exact if it has only one decision value; otherwise, it is rough. The union of all the exact granules with respect to C is referred to as the positive region, described as $POS_C\ (D) = \{[x]_C || I_d([x]_C)| = 1\}$. Based on the indiscernibility relationship and the positive region, a discernibility matrix is defined as follows.

Definition 2. *Given an information table S, a discernibility matrix M based on the positive region is defined as*

$$m(i,j) = \begin{cases} \{a \in C | I_a(x_i) \neq I_a(x_j)\}, I_d(x_i) \neq I_d(x_j) \wedge (x_i \in POS_C\ (D) \vee x_j \in POS_C\ (D)) \\ \varnothing, \qquad \text{else} \end{cases} \quad (2)$$

Definition 3. *Given an information table S, an attribute set $R \subseteq At$ is called a reduct if it satisfies the following two conditions [7]:*

(1) $IND(R) = IND(At)$;
(2) $\forall a \in R, IND(R - \{a\}) \neq IND(At)$.

If a discernibility matrix $M = \{m(i,j)\}$ is constructed, then the two conditions mentioned above can be described as follows:

(1) $\forall m(i,j) \neq \varnothing \Rightarrow R \cap m(i,j) \neq \varnothing$;
(2) $\forall a \in R, \exists m(i,j) \neq \varnothing \wedge m(i,j) \cap (R - \{a\}) = \varnothing$.

A reduct is a subset of attributes that is "jointly sufficient and individually necessary" to represent the knowledge equivalent to the attribute set C. In general, an information table may have multiple reducts. The set of these reducts is denoted as $RED(S)$, and the intersection of all reducts is the core set, $Core(S) = \cap RED(S)$ or $Core(M) = \cup_{|m(i,j)|=1} m(i,\mathrm{j})$. If an attribute subset R' only satisfies the first condition, then it is referred to as a super reduct, i.e., $\exists R \in RED(S),\ R \subseteq R'$.

3. Recursive Attribute Reduction Algorithm Based on Priority Sequence

In this section, a priority-optimal reduct is defined first. Subsequently, a heuristic approach is discussed, and an attribute reduction algorithm based on a priority sequence is proposed using recursion.

3.1. Priority Sequence and Priority-Optimal Reduct

In this paper, prior knowledge of a blast furnace is treated as a priority sequence of attributes. For a priority sequence $\{c_1, c_2, \ldots, c_n\}$, attribute c_1 has the highest priority, and c_n has the lowest priority. In practical applications, we can determine this sequence based on empirical knowledge or mechanistic analysis results. In other words, when assigning numbers to these attributes, those deemed more important are placed at the front. The priority-optimal reduct is thus defined as follows.

Definition 4. *Given a discernibility matrix M and a priority sequence $\{c_1, c_2, \ldots, c_n\}$, a reduct $R \in RED(M)$ is called the priority-optimal reduct (POR) if $\forall R' \in RED(M) - R, \exists c \in R - (R \cap R')$, so that $p(c) > p(c')$, where $p(c)$ is the priority of attribute c, and $c' \in R' - (R \cap R')$.*

Based on Definition 4, it is straightforward to conclude that POR is unique for a given priority sequence. In other words, each priority sequence can accurately map a corresponding POR. Hence, if the priority sequence represents prior knowledge, then POR is the optimal solution.

However, even though the priority sequence is taken into account, it is still difficult for the traditional attribute reduction algorithm (shown in Algorithm 1) to obtain the POR. A typical example is shown in Example 1.

Example 1. *Suppose a discernibility matrix M has four non-empty elements: $\{a,b\}$, $\{a,c\}$, $\{b,d\}$, and $\{c,d\}$.*

For a priority sequence $\{a,b,c,d\}$, it obtains $R' = \{a,b,c\}$ after step 3, and the attribute "a" is removed in step 4, yielding the reduct $\{b,c\}$. However, the matched POR is $\{a,d\}$ because attribute "a" has a higher priority.

Example 1 illustrates that the traditional attribute reduction algorithm is not suitable for calculating POR because high-priority attributes may be removed during the reduction process. In other words, a combination of lower-priority attributes replaces an attribute with a higher priority. Therefore, to obtain POR, new approaches are necessary to avoid this outcome.

3.2. Calculation Method on POR

For a heuristic reduction algorithm, each condition attribute is evaluated in turn according to the given priority sequence $\{c_1, c_2, \ldots, c_n\}$. The key challenge is to determine whether the evaluated condition attribute belongs to POR. Next, we analyze how to identify the first attribute of POR and the remaining attributes, respectively.

3.2.1. Calculation of the First Attribute of POR

Theorem 1. *For a priority sequence $\{c_1, c_2, \ldots, c_n\}$, if $\exists R \in RED(M)$ so that $c_1 \in R$, then $c_1 \in POR$.*

Proof. For $\forall R' \in RED(M)$, if $\{c_1\} \cap R' = \varnothing$, it has $p(c_1) > max(p(c'))$, where $c' \in R' - (R \cap R')$. According to Definition 4, R' is not the POR. In other words, POR includes attribute c_1. □

According to Theorem 1, the first attribute of POR can be found using the following approach.

Approach 1. *Attribute c_i is the first attribute of POR if it satisfies the following two conditions:*
$\exists R \in RED(M), c_i \in R$;
$\forall c \in C,\ if\ p(c) > p(c_i), then\ \{c\} \cap (\cup RED(M)) = \varnothing$.

However, calculating all the reducts $RED(M)$ is impractical in a heuristic algorithm. Thus, the notions of free matrix and absolute redundant attribute set are proposed in this paper to identify the first attribute of POR.

Definition 5. *The absolute redundant attribute set of a decision table S is defined as*

$$ARAS(S) = \{\mathrm{c} \in \mathrm{C} | \forall \mathrm{R} \in \mathrm{RED(S)}, \mathrm{c} \notin \mathrm{R}\} \tag{3}$$

$ARAS(S)$ is also referred to as $ARAS(M)$ if a discernibility matrix M is considered. Based on Definition 5, one can derive the following properties:

(1) $Core(M) \cap ARAS(M) = \varnothing$;
(2) $\forall c \in C$, *if* $c \notin ARAS(M)$, *then* $\exists R \in RED(M)$, $c \in R$;
(3) $ARAS(M) \cap POR = \varnothing$.

Theorem 2. *Given a discernibility matrix M and an attribute priority sequence $\{c_1, c_2, \ldots, c_n\}$, if $ARAS(M) = \varnothing$, then c_1 is the first attribute of POR.*

Proof. Based on Definitions 4 and 5, since $ARAS(M) = \varnothing$, $\exists R \in RED(M), c_1 \in R$. Considering that c_1 has the highest priority, c_1 must belong to POR, and it is its first attribute. □

If $ARAS(M) \neq \varnothing$, one can remove $ARAS(M)$ from the discernibility matrix to calculate the first attribute of POR. The resulting discernibility matrix is referred to as a free matrix, defined as follows.

Definition 6. *Discernibility matrix M is a free matrix if, for any non-empty elements $m(x,y), m(x',y') \in M$, it holds that $m(x,y) \not\subset m(x',y')$.*

The free matrix M does not contain any absolute redundant attribute, and the relevant analysis is presented below.

Theorem 3. *Given discernibility matrix M, if $\forall m(x,y), m(x',y') \in M$, $m(x,y) \not\subset m(x',y')$, then $ARAS(M) = \varnothing$.*

Proof. For any attribute $c \in C$, if c is a core attribute, then $c \notin ARAS(M)$. If attribute c is not a core attribute, let $M = M_c + M_{c}\prime$, where $M_c = \{m(x,y) \in M | c \in m(x,y)\}$, $M_c\prime = \{m(x,y) \in M | c \notin m(x,y)\}$. Select an element $m(p,q) \in M_c$ and define $A = m(p,q) - \{c\}$. One can construct a new discernibility matrix $M\prime = \{m\prime(x,y) | m\prime(x,y) = m(x,y) - A\}$. Matrix M' clearly has the following features: (1) attribute c is a core attribute of M' and (2) under the existing condition $(m(x,y) \not\subset m(x',y'))$, $|m(x,y) \cup m(x',y') - (m(x,y) \cap m(x',y'))| \geq 2$. Thus, for any non-empty element $m(x,y) \in M$, $m\prime(x,y) \neq \varnothing$. Hence, based on Definition 3, $C - A$ is a super reduct of M and $\forall R' \in RED(M\prime)$, $R' \in RED(M)$. Since $c \in CORE(M\prime)$, there exists a reduct R' that includes c. Therefore, attribute c is not a redundant attribute in R', and $ARAS(M) = \varnothing$. □

Based on Theorems 2 and 3, we obtain an important approach.

Approach 2. *The highest priority attribute of a free matrix is the first attribute of POR.*

The free matrix is constructed by Algorithm 2.

Algorithm 2. Construct the free matrix.

Input: Discernibility matrix M.
Output: The related free matrix.
Step 1: Sort all the non-empty elements $m(x,y)$ by $|m(x,y)|$; let num be the number of non-empty elements.
Step 2:
Set $i = 1$
  While $i \leq num - 1$, **do**
$j = i + 1$;
While $j \leq num$, **do**
If $m_i(x,y) \subset m_j(x,y)$ **then**
        delete $m_j(x,y)$ from $M\prime$, $num = num - 1$
      Else
        $j = j + 1$.
      End
    End
        $i = i + 1$
  End
Step 3: Output the free matrix.

3.2.2. Calculation on Other Attributes of POR

Besides the first attribute, the other attributes of POR are checked according to Theorem 4.

Theorem 4. *For attribute c_{k+1}, if $\exists R \in RED(M)$, $B_k \cup \{c_{k+1}\} \subseteq R$, then $c_{k+1} \in POR$, where $B_k = POR \cap A_k$ and $A_k = \{c_1, c_2, \ldots, c_k\}$.*

Proof. Let $P = POR - B_k$. According to the definition of B_k, the attribute set P is a subset of $C - A_k$. Since there exists a reduct R that includes B_k and attribute c_{k+1}, B_k cannot discern all elements of matrix M, i.e., $P \neq \varnothing$. Suppose $c_{k+1} \notin POR$. In that case, the reduct R is not POR. However, $\forall c \in P$, $c_{k+1} \in R - B_k$ satisfies $p(c_{k+1}) > p(c)$. This conflicts with Definition 4. Hence, the above assumption is invalid, implying $c_{k+1} \in POR$. □

According to Theorem 4, the conclusion is that the key step is to verify whether there exists a reduct that contains both c_{k+1} and the attribute set $B_k = POR \cap \{c_1, c_2, \ldots, c_k\}$.

Considering that a reduct should be "jointly effective and individually necessary," we introduce the notion of the necessary element set to represent "individually necessary."

Definition 7. *Given a discernibility matrix $M = \{m(x, y)\}$, the necessary element set (NES) of attribute c with respect to attribute set B is $NES_B(c) = \{m(x, y) | m(x, y) \cap B = \{c\}\}$.*

The notion of NES is similar to that of a core set, since both are related to "necessary". The core set represents the attributes that are necessary for all reducts. Meanwhile, $NES_B(c)$ indicates whether attribute c is necessary to attribute set B. It has the following:

(1) $CORE(M) = \{c \in C | NES_C(c) \neq \varnothing\}$;
(2) If $\{c\} \subseteq A_1 \subseteq A_2$, then $NES_{A_2}(c) \subseteq NES_{A_1}(c)$.

If $NES_B(c) \neq \varnothing$, then the attribute set $B - \{c\}$ is not a reduct because it cannot discern the non-empty elements in $NES_B(c)$. Conversely, if $NES_B(c) = \varnothing$, then $\forall m(x, y) \cap \{c\} \neq \varnothing$, $m(x, y) \cap (B - \{c\}) \neq \varnothing$. This means that all elements discerned by attribute c are also discerned by $B - \{c\}$. Thus, attribute set B is not a reduct.

Based on the above analysis, we redefine the reduct using NES.

Definition 8. *An attribute set R is called a reduct if and only if it satisfies the following conditions:*

(1) For each non-empty element $m(x, y) \in M$, $m(x, y) \cap R \neq \varnothing$;
(2) For each attribute $c \in R$, $NES_R(c) \neq \varnothing$.

Based on Definition 8, we have the following conclusion about $NES_B(c)$.

Theorem 5. *Given a discernibility matrix $M = \{m(x, y)\}$ and an attribute set $B \subseteq C$, if there exists a reduct $R \in RED(M)$ that includes the attribute set B, then $\forall c \in B, NES_B(c) \neq \varnothing$.*

Proof. Based on Definition 8, R satisfies $\forall c \in R$, $NES_R(c) \neq \varnothing$. Because $B \subseteq R$, we have $\forall c \in B, NES_B(c) \neq \varnothing$ since $NES_R(c) \subseteq NES_B(c)$. □

According to Theorem 5, if $\exists c \in B$, $NES_B(c) = \varnothing$, then no reduct can include B.

Based on the above discussions, we propose a recursive algorithm to determine the other attributes of POR. Recursion is a kind of self-relation and is represented as a function capable of calling itself within a program. Once a recursive function calls itself, it reduces a problem into sub-problems. The recursive call process continues until it reaches an end point where the sub-problem cannot be reduced further. Thus, there are two elements in a recursive method: simple repeated logic and a termination condition. In this paper, an additional border logic is also adopted to ensure that the result is a reduct.

Simple Repeated Logic

Repeated logic refers to the similarity between a problem and its sub-problems, which constitutes the main part of recursion.

In this method, repeated logic is utilized to verify the existence of a reduct that includes $B_k \cup \{c_{k+1}\}$, and it can be divided into two parts: (1) calculation NES and (2) judgment logic. The NES calculation of $B_k \cup \{c_{k+1}\}$ is based on Definition 7, and further descriptions are unnecessary. For the judgment logic, there are two cases.

(1) If $\forall c \in B_k \cup \{c_{k+1}\}, NES_{B_k+\{c_{k+1}\}}(c) \neq \varnothing$, then we accept c_{k+1}, $B_{k+1} = B_k \cup \{c_{k+1}\}$, and we call this method with the updated parameter set.

(2) If $\exists c \in B_k \cup \{c_{k+1}\}, NES_{B_k \cup \{c_{k+1}\}}(c) = \varnothing$, then we reject c_{k+1}, $B_{k+1} = B_k$, and we remove c_{k+1} from the discernibility matrix.

A simple example is described as follows.

Example 2. *Given* $M = \{\{a,b\},\ \{a,c\},\ \{b,d\},\ \{c,d\}\}$ *and the priority sequence* $\{a,b,c,d\}$, M *is a free matrix.*

First, according to Approach 2, attribute a is the first attribute of POR. Next, $B_1 = \{a\}$ and the other attributes are evaluated in turn.

For attribute b, $NES_{B_1 \cup \{b\}}(a) = \{\{a,c\}\}$, $NES_{B_1 \cup \{b\}}(b) = \{\{b,d\}\}$. Attribute b is accepted, and the attribute set becomes $B_2 = \{a,b\}$.

For attribute c, $B_2 = \{a,b\}$, $NES_{B_2 \cup \{c\}}(a) = \varnothing$, $NES_{B_2 \cup \{c\}}(b) = \{\{b,d\}\}$, $NES_{B_2+\{c\}}(c) = \{\{c,d\}\}$. Since $NES_{B_2 \cup \{c\}}(a) = \varnothing$, attribute c is rejected and the new matrix is $M_1 = \{\{a,b\},\ \{a\},\ \{b,d\},\ \{d\}\}$, $B_3 = B_2 = \{a,b\}$.

Additional Border Logic

Since the logic above uses only a necessary condition to test attributes in priority sequence order, we will encounter a border during recursion, referred to here as core conflict.

Definition 9. *Given a discernibility matrix* M*, attribute set* B*, and a core attribute* $c' \in CORE(M)$*, the core conflict between* B *and* $c'(c' \notin B)$ *is described as follows:* $\exists c \in B$, $NES_B(c) \neq \varnothing$, $NES_{B \cup \{c'\}}(c) = \varnothing$.

Since $c' \in CORE(M)$, c' must belong to all reducts. However, due to this core conflict, no reduct can include both B and c'. Thus, we present the following approach.

Approach 3. *Given a discernibility matrix* M *and an attribute set* B*, if there is a core conflict between* B *and a core attribute, then no reduct can include* B.

At the start of this method, no core attribute exists in the free matrix M because of step 1 of Algorithm 1. As recursion proceeds, some attributes are removed from M, which causes certain attributes to become core attributes in the new discernibility matrix M_k. If these core attributes conflict with higher-priority attributes, a core conflict will arise in the recursion.

The occurrence of core conflicts implies that at least one redundant attribute is present. The simplest way to address this is to remove any attributes for which $NES_{B \cup \{c'\}}(c) = \varnothing$ from the related discernibility matrix. By doing so, we eventually obtain a reduct. However, this reduct may not be POR. A simple example is provided in Example 3.

Example 3. *Given* $M = \{\{a,b\},\ \{a,e\},\{b,d\},\{c,e\},\ \{d,e\}\}$ *and the priority sequence* $\{a,b,c,d,e\}$, M *is a free matrix, and attribute* a *is the first attribute of POR. Using repeated* *logic, we obtain* $B_3 = \{a, b, c\}$, $NES_{B_3}(a) = \{\{a, e\}\}$, $NES_{B_3}(b) = \{\{b, d\}\}$, $NES_{B_3}(c) = \{\{c, e\}\}$. *For the next attribute d, we find* $NES_{B_3 \cup \{d\}}(b) = \varnothing$, *and d is not a core attribute. Attribute d is then removed from M, producing* $M_1 = \{\{a, b\}, \{a, e\}, \{b\}, \{c, e\}, \{e\}\}$. *For the last attribute e,* $NES_{B_3 \cup \{e\}}(a) = \varnothing$, $NES_{B_3 \cup \{e\}}(b) = \{\{b\}\}$, $NES_{B_3 \cup \{e\}}(c) = \varnothing$. *Since e is a core attribute, it cannot be removed. In this case, the core attribute e conflicts with two higher-priority attributes a and c. If the conflict is ignored, the attribute set* $\{b, e\}$ *would be obtained by removing a and c. Clearly,* $\{b, e\}$ *is not POR, as there exists another reduct* $\{a, c, d\}$ *in which attribute a has higher priority than any of the attributes in* $\{b, e\}$.

Thus, the recursive process must stop if a core conflict arises, i.e., the method reaches a wrong border and must backtrack. This example illustrates the crucial role of core conflict. It helps the algorithm identify situations where accepting the current attribute would conflict with higher-priority attributes. Hence, an additional border logic is proposed to address this type of problem.

The additional border logic identifies the last attribute that can be removed from B_k, allowing the method to restart from a new start point. Because the border logic operates at multiple levels of recursion, it can be split into two parts: (1) judgment of core conflict in the current level and (2) handling the return operation in the subsequent level.

There are three cases:

Case 1. The current attribute is not in B_k (rejected by Simple Repeated Logic in the previous step); continue the return operation.

Case 2. The current attribute is in B_k, but it is a core attribute that cannot be deleted; continue the return operation.

Case 3. The current attribute is in B_k, but it is not a core attribute; then, refuse it, remove it from M, and enter the next recursion from the new start point.

The Terminate Condition

The terminate condition marks the endpoint of the recursion and is triggered once all attributes in the priority sequence have been tested. If the core conflict is treated as an unsuccessful border, then the terminate condition can be regarded as a successful border. In practical programming, we use a flag to distinguish between the two situations.

3.2.3. The Complete Reduction Algorithm Based on Recursion

Based on the above discussions, the complete algorithm is described as shown in Algorithms 3 and 4. In Algorithm 3, we provide the framework of the complete attribute reduction algorithm, and the details of its recursion function are described in Algorithm 4.

Algorithm 3. Reduction algorithm based on recursion for calculating *POR*.

Input: Information table S and attribute priority sequence $\{c_1, c_2, \ldots, c_n\}$.
Output: The corresponding *POR*.
Step 1: Construct the discernibility matrix M, calculate the core attribute set $CORE(M)$, and delete the elements from M that can be discerned by $CORE(M)$.
Step 2: Construct the free matrix $M\prime$ and delete attributes in the attribute priority sequence that do not appear in the free matrix ($Core(M)$ and $ARAS(M)$). The new attribute priority sequence P is $\{c_{i1}, c_{i2}, \ldots, c_{im}\}$, and the attribute with the highest priority in $M\prime$ is c_{i1}, i.e., $B_1 = \{c_{i1}\}$.
Step 3: Use Algorithm 4 to test other attributes in P with the input ($B = B_1, P, M = M', k = 2$), and output ($flag, B\prime$).
Step 4: Output $POR = B\prime \cup CORE(M)$.

Algorithm 4. Recursive function f.

Input: Attribute set B, priority sequence P, discernibility matrix M, k.
Output: Attribute set $B\prime$, $flag$.
Step 1: If $k > |P|$ **then**
Return with $flag = 1, B' = B$//Valid reduct found, recursion ends
End
Step 2: For each attribute c in $B \cup \{c_{ik}\}$, calculate $NES_{B\cup\{c_{ik}\}}(c)$ based on M.
Step 3: If $\forall c \in B \cup \{c_{ik}\}$, $NES_{B\cup\{c_{ik}\}}(c) \neq \varnothing$ **then**
Call f with input $(B \cup \{c_{ik}\}, P, M, k+1)$//Accept c_{ik}, recursive call
Else
Refuse c_{ik} and test if c_{ik} is a core set of M
If c_{ik} is a core set, then
Return with $flag = 0$//Core conflict detected, backtrack
Else
Delete c_{ik} from M
Call f with input $(B, P, M, k+1)$//Recursive call without c_{ik}
End
End
Step 3: Process the result returned:
If $flag = 0 \ \wedge \ c_{ik}$ not be rejected $\wedge$ c_{ik} not a core of M
Refuse c_{ik} and delete c_{ik} from M
Call f with input $(B, P, M, k+1)$ and return the result returned directly
Else
Return the result returned directly
End

A simple proof of Algorithm 3 is as follows.

Since we evaluate each attribute using the priority sequence, and a high-priority attribute is deleted only if no reduct includes the chosen attribute set, the main problem is to prove that the selected attribute set is a reduct. First, due to the repeated logic, the finally obtained attribute set $B\prime$ must satisfy $\forall c \in B\prime, NES_{B'}(c) \neq \varnothing$. Second, suppose $B\prime$ is not a super reduct of $M\prime$; then, there exists at least one element that cannot be discerned by the attribute set. Owing to the deletion mechanism in the repeated logic, any element that cannot be discerned will lead to a core conflict, which prevents such an element from remaining. Thus, the attribute set obtained by Algorithm 3 is exactly the POR.

We also illustrate the complete calculation process in Example 3 in Figure 1.

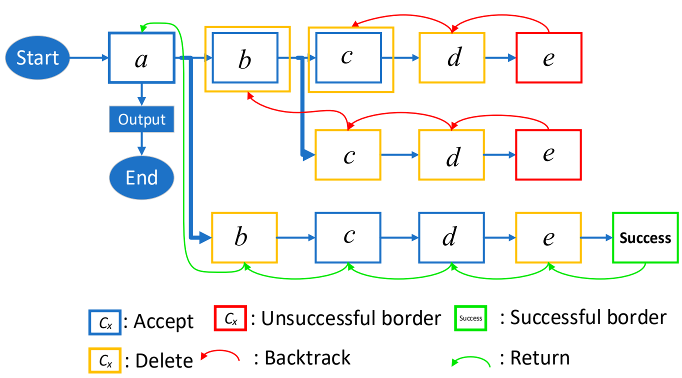


Figure 1. Calculation in Example 3.

First, attributes a, b, and c are accepted by the repeated logic, whereas d is rejected. The new discernibility matrix is denoted by M_1.

$$M_1 = \{\{a,b\},\ \{a,e\},\ \{b\},\ \{c,e\},\ \{e\}\}$$

It is observed that attribute e is a core attribute of M_1 and conflicts with $\{a,\ c\}$. Hence, the algorithm reaches an unsuccessful border. Then, the algorithm gradually backtracks using the additional border logic. First, attribute d satisfies Case 1, so the algorithm continues the return operation. Next, attribute c satisfies Case 3 and is deleted because it is not a core attribute.

Now, the algorithm moves to the next recursion level from the new start point (attribute d), where $B_2 = \{a,b\}$, $M_2 = \{\{a,b\},\ \{a,e\},\ \{b,d\}, \{e\},\ \{d,e\}\}$. Attribute d is rejected by the repeated logic because $NES_{B_2 \cup \{d\}}(b) = \varnothing$. After removing attribute d, the discernibility matrix becomes $M_3 = \{\{a,b\},\ \{a,e\},\ \{b\},\ \{e\},\ \{e\}\}$. It is determined that attribute e is a core attribute and conflicts with $\{a,b\}$. Accordingly, the algorithm gradually backtracks to the previous level where the last accepted attribute was b.

Third, attribute b is deleted and the other attributes are checked in order until a successful border appears. Eventually, the output POR is $\{a,c,d\}$.

We also compared the proposed algorithm with a classical heuristic algorithm, a recently reported algorithm from the literature [7], and the state-of-the-art distributed attribute reduction algorithm RA-MRS described in [3]. The related experimental results are listed below.

These tested datasets originate from UCI and are uniformly discretized if they have continuous attribute values. For example, Sonar_16 indicates that the Sonar dataset's continuous attribute values are uniformly discretized into 16 intervals.

In Table 1, underscores highlight the differences from other algorithms, and numbers represent the index of attributes. The experimental results demonstrate that our algorithm effectively identifies the priority-optimal reduct, while the compared algorithm does not necessarily. Moreover, the reduction set from our algorithm is generally larger than those of the compared algorithms. That is, it is difficult for our algorithm to obtain a minimum reduct because data backgrounds are taken into consideration and some high-priority attributes must remain.

Table 1. Experimental results.

Dataset	Classical Algorithm	The Proposed Algorithm	Reduction Algorithm in [7]	RA-MRS in [3]
Sonar_2	5,7,11,16,17,2022,24,26,27, 30,33,35,37,53,54	**1**,5,8,13,16,17,19–22,26,27, 30–34,36, 37, 42,43, 53, 54	5,7,11,16,17,20,21,22,24,26, 27,30–33, 35, 37,53,54	1–30
Sonar_4	1,6–8,11,12,17, 18, 21, 23	**1–6**,23,25,31, 35, 41,43,47	1,6–8,11,12,17, 18,21, 23	1–23
Sonar_8	1,2,7,9,10,14,18	**1–7**,44,50,58	1,2,7,9,10,14,18	1–12
Sonar_16	1–5,9	1–5,**7**,53	1–5,9	1–8
Iono_2	1,3,5–7,9,11–14, 16,19,23,25,29,30,33	**1–3**,5,7,9,11–17,19,23,25, 29, 32, 33	1,3,5–7,9,11–14, 16,19,23,25,29,30,33	1–32
Iono_4	2–5,7,9,10,15, 23, 24	**1–7**,23,24,26, 33	2–5,7,9,10,15, 23,24	1–24
Iono_8	1–4,6,7,9	**1–5**,13,16,29	1–4,6,7,9	1–10
Iono_16	2,3,5,7,8	**1–4**,6,9,28	2,3,5,7,8	1–8
Zoo	3,4,6,8,13	**1**,3,6,7,10,12,13	3,4,6,8,13	1–12
Wine_2	1–12	1–12	1–12	1–12
Wine_4	1–7,9	1–7,9	1–7,9	1–12
Wine_8	1–3,5,6	1–4,8	1–3,5,6	1–12

Table 1. *Cont.*

Dataset	Classical Algorithm	The Proposed Algorithm	Reduction Algorithm in [7]	RA-MRS in [3]
Wine_16	1–4.	1–4.	1–4	1–7
Fertility_4	1–3,5–7,9	1–3,5–7,9	1–3,5–7,9	1–8
Fertility_8	1,2,4–7	**1–3**,7–9	1,2,4–7	1–7
Fertility_16	1–4,6,7	1–4,6,7	1–4,6,7	1–7

4. Application in Blast Furnace Smelting

Blast furnace smelting is a complex, nonlinear, and high-dimensional dynamic process, as shown in Figure 2. Raw materials such as iron and coke are fed from the top. As they move downward, oxygen-enriched hot air and pulverized coal are conveyed from the bottom of the blast furnace and eventually flow upward. Complex reactions of various materials occur in multi-phase states while a variety of physical changes and chemical reactions occur simultaneously during the two-directional motion [16].

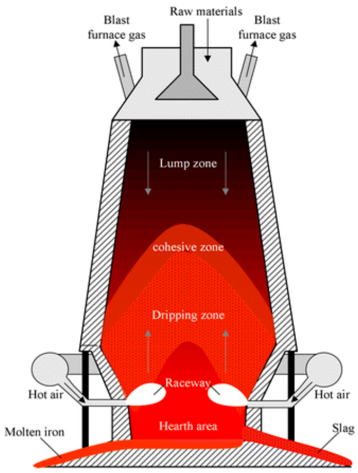


Figure 2. Blast furnace structural diagram.

The main behavior of silicon in the blast furnace smelting process is the reduction reaction. First, during the SiO_2 reduction by coke carbon or carbon dissolved in hot metal, part of the silicon mixes with hot metal in a liquid phase as follows:

$$SiO_2 + 2C = Si + 2CO,$$

Meanwhile, most of the silicon transforms to gaseous SiO by the following reaction:

$$SiO_2 + C = SiO + CO.$$

SiO then rises with blast furnace gas and is dissolved by both slag and hot metal from the cohesive zone. The dissolved SiO reacts again with the coke in metal:

$$SiO + C = Si + CO.$$

The thermal state of the blast furnace is one of the most important factors, as temperature significantly influences reduction reactions. Due to the enclosed nature of blast furnaces, directly obtaining the thermal state poses a challenge. Therefore, the silicon content is used as an indicator to determine the blast furnace state. Predicting silicon content has also become a focus of study, and many machine learning models—such as support vector regression and neural networks—have been applied to this task. One of the most important factors of machine learning performance is data quality. Thus, selecting an optimal feature set is essential for providing high-quality inputs.

In this section, we first determine the attribute priority sequence related to silicon content. Next, the reduction algorithm based on priority sequence is applied for feature selection. Finally, a long short-term memory recurrent neural network (LSTM-RNN) is employed to predict silicon content and verify the validity of POR.

4.1. Data Description and Priority Sequence

Data were collected from the No. 2 blast furnace of Liuzhou Steel in China, which has a volume of 2650 m^3. A total of 1200 data groups are available to validate the method; 800 of them are used as the training set and 400 as the test set. In the dataset related to the hot metal silicon content, there are sixteen condition attributes and one decision attribute.

The following principles are followed when determining the attribute priority sequence.

(1) The mechanisms of blast furnace smelting should be considered first. For example, since blast furnace smelting is a continuous process, the silicon content at the last time point strongly influences the current silicon content.
(2) We also considered the staff's experience, since they know which attributes carry the greatest importance during operation.
(3) Correlation analysis between the condition attributes can serve as a reference for the attribute priority sequence.

Considering the above factors comprehensively, we obtain the following attribute priority sequence: {latest silicon content, theoretical burning temperature, bosh gas index, bosh gas volume, actual wind speed, standard wind speed, gas permeability, blast momentum, furnace top pressure, oxygen enrichment percentage, cold wind flow, hot blast temperature, pressure difference, hot blast pressure, cold wind pressure, oxygen enrichment pressure}. For convenience, we use c_1–c_{16} to represent these condition attributes, and the subscript of a symbol represents its position in the priority sequence, i.e., its priority.

4.2. Attribute Reduction

In our work, we uniformly discretized the data for every attribute into 10 intervals. We then ran the proposed attribute reduction procedure as follows.

First, we constructed the discernibility matrix M based on Definition 2. Then, we calculated the core attribute set $CORE(M) = \{c_1, c_2, c_7, c_{10}, c_{12}, c_{16}\}$ and deleted the elements that can be discerned by $CORE(M)$ from M. Next, the free matrix was constructed and we obtained $ARAS(M) = \varnothing$. Thus, the new attribute priority sequence is $\{c_3, c_4, c_5, c_6, c_8, c_9, c_{11}, c_{13}, c_{14}, c_{15}\}$, with c_3 as the highest-priority attribute. Finally, Algorithm 3 was executed. The specific calculation process is shown in Figure 3.

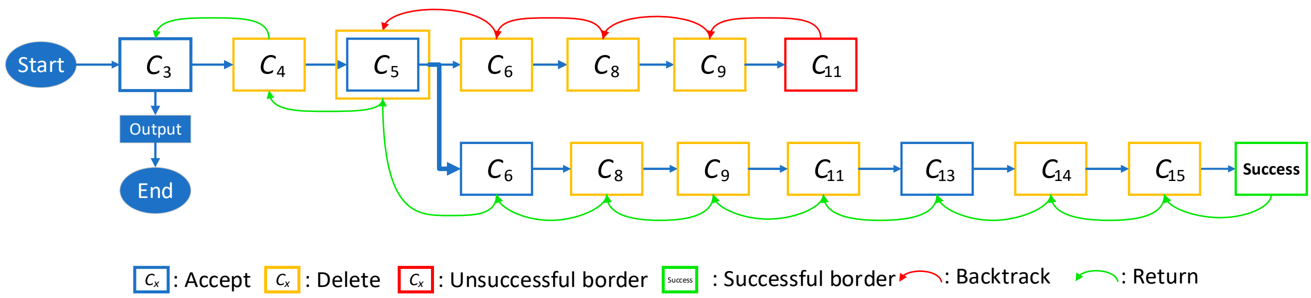


Figure 3. Recursive process.

According to the simple repeated logic, we added c_3; deleted c_4; added c_5; and deleted c_6, c_8, c_9. Then, it was determined that c_{11} is a core attribute that conflicts with $\{c_3, c_5\}$. This meant that the algorithm reached an unsuccessful border. Next, this algorithm gradually backtracked to the level where the last attribute c_5 was accepted. In the following steps, c_5 was removed and the attributes left were checked in order until a successful border was reached. Eventually, we obtained $\{c_3, c_6, c_{13}\}$ and generated $POR = CORE(M) \cup \{c_3, c_6, c_{13}\}$.

To demonstrate the effectiveness of our algorithm, we compared it to the traditional addition–deletion algorithm, as shown in Algorithm 1. The results are presented in Table 2.

Table 2. Results of attribute reduction.

Algorithm	Reduct
Proposed algorithm	$c_1, c_2, c_3, c_6, c_7, c_{10}, c_{12}, c_{13}, c_{16}$
Addition–deletion algorithm	$c_1, c_2, c_6, c_7, c_8, c_{10}, c_{12}, c_{16}$

For ease of description, we denote the reduct obtained by the proposed recursion algorithm as R_r, and the other as R_a. From Table 2, with the exception of the attributes both algorithms share, c_3 holds the highest priority in R_r, whereas c_8 does so in R_a. Since c_3 is much more important than c_8 based on the priority sequence, the reduct R_r is more aligned with the data-driven priority sequence. These reducts also show that the classic addition–deletion algorithm is less effective in calculating the priority-optimal reduct. By retaining those high-priority attributes, our reduction sets, while slightly larger in the number of attributes, perform better in capturing domain knowledge.

4.3. Prediction with LSTM-RNN

A blast furnace functions as a dynamic delay system, in which its current state depends on its previous state. LSTM-RNN is a gated recurrent neural network whose structure is shown in Figure 4. Each output of LSTM-RNN is also related to the previous state. Moreover, LSTM-RNN can selectively use previous state information to predict the current state based on the input, which makes it more flexible and suitable for the prediction of hot metal silicon content.

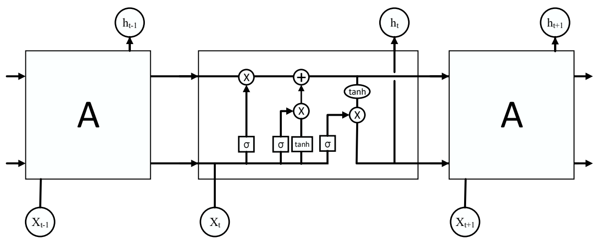


Figure 4. Structure of LSTM-RNN.

Considering the above discussion and the complexity of blast furnace smelting, we adopted LSTM-RNN to predict the hot metal silicon content. The actual model consists of one LSTM layer with a 10-dimensional output, and one output layer with a 1-dimensional output. Additional parameter settings of the LSTM layer are shown in Table 3 (the deep learning framework used is Keras 2).

To ensure a valid comparison, we adjust the validation set five times during the training process. The validation set is a continuous part of the training set, and the remaining part of the training set is used to train the neural network. The results are shown in Table 4 (tests with the same test number share the same validation set). MSE and Hit are calculated as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2,$$

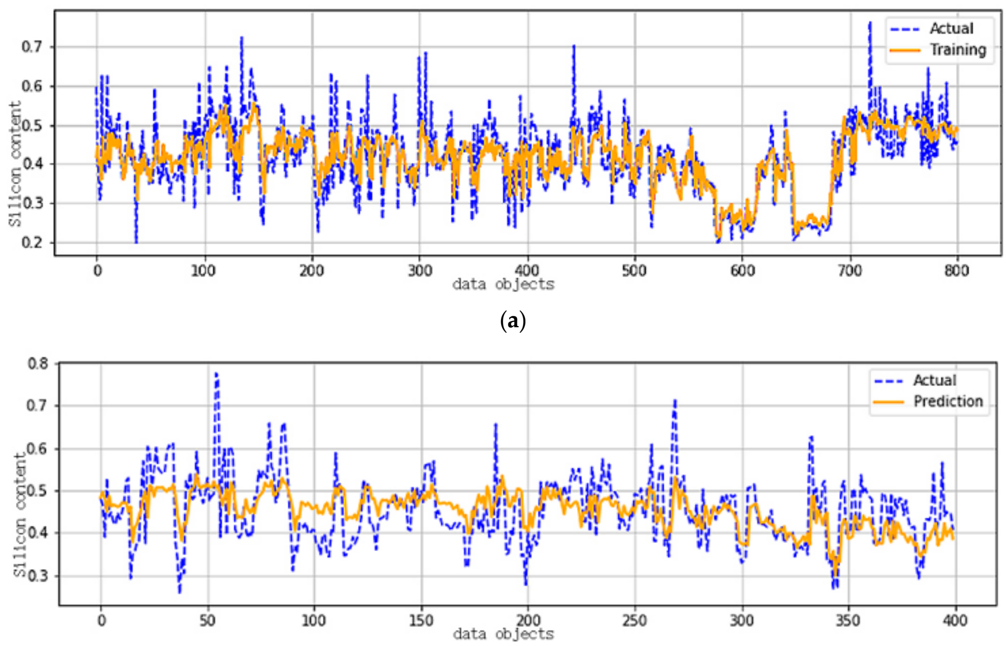
$$Hit = \frac{100\%}{n} |\{\hat{y}_i || \hat{y}_i - y_i| \leq 0.1\}|.$$

Table 3. Parameter settings of LSTM.

Parameter	Setting
Activation	relu
Timesteps	1
input_dim	| reduct |
batch_size	100
Others	default

Table 4. Prediction results.

Reduct	Test Number	Train_MSE	Test_MSE	Hit (Train)	Hit (Test)
R_r	1	0.0042	0.0051	88.75%	85.25%
	2	0.0042	0.0048	88.35%	86.75%
	3 (Figure 4)	0.0043	0.0047	87.93%	87.75%
	4	0.0042	0.0053	88.49%	84.25%
	5	0.0046	0.0047	86.55%	87.50%
R_a	1	0.0042	0.0055	88.06%	83.25%
	2	0.0040	0.0054	89.17%	83.50%
	3 (Figure 5)	0.0042	0.0057	88.89%	82.25%
	4	0.0045	0.0057	86.67%	81.50%
	5	0.0048	0.0052	87.36%	84.25%



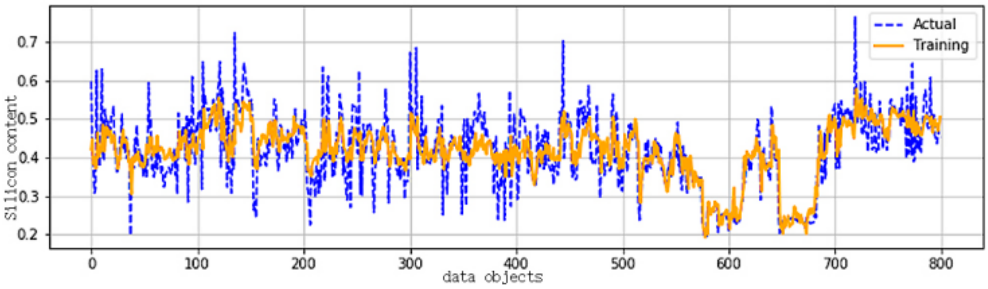
(**a**)

(**b**)

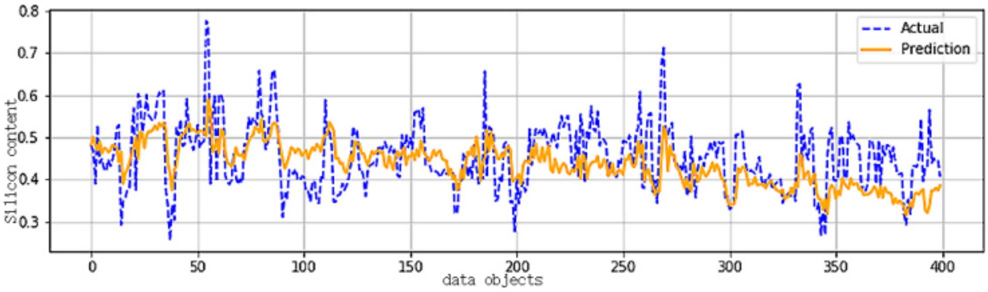
Figure 5. Predictive results based on R_r.(**a**) Training set. (**b**) Test set.

From Table 4, models trained with R_r achieve better performance than those trained with R_a. Specifically, the training set results are almost the same (MSE about 0.043, Hit about 88%). However, the situation is different for the test set. The average Test_MSE and Hit for R_r are 0.0049 and 86.3%, respectively, while 0.0055 and 82.9% for R_a. Compared with R_a, the Test_MSE decreases by 10.9% and Hit increases by 4.1%.

For further analysis, we selected the most representative test, i.e., test number 3, as shown in Figures 5 and 6, for observation and comparison. For the training set, the figures are almost the same. However, for the test set, models trained based on R_a fail to track the change in silicon content once the time point exceeds 300, whereas the models trained based on R_r exhibit superior performance and effectively capture the trend of silicon content.



(**a**)



(**b**)

Figure 6. Predictive results based on R_a. (**a**) Training set. (**b**) Test set.

Through the above analysis, it is shown that the priority-optimal reduct retains more precise and relevant information than the classical reduction sets, and the related models demonstrate stronger generalization abilities. Therefore, the attribute reduction algorithm based on recursion proves to be practical in real-world applications.

5. Conclusions

In this study, we introduced a new definition of the priority-optimal reduct for complex industrial processes within rough set theory. Based on this, a recursive attribute reduction algorithm was developed. As the first recursive construction in the history of rough sets, it has important research value. Moreover, the results of experiments on silicon content prediction in a blast furnace demonstrate the effectiveness of our algorithm under complex blast furnace conditions.

Our work successfully applied the new attribute reduction to the feature selection of hot metal silicon content data from the blast furnace. In addition to the description of prior knowledge, the characteristics of the data itself should also be considered. Since the data are numerical, a discernibility relation that relies on discrete data may introduce quantization error. Thus, further investigation of tolerance relations, fuzzy relations, or a new discernibility relation is expected to yield better performance for this problem.

It should be noted that the performance of the proposed algorithm in practical applications heavily depends on a reasonable priority sequence, which is usually derived from experiential knowledge or mechanism analysis results. Consequently, there exists a potential overfitting risk when applying it to specific domain datasets. We leave a more robust priority sequence determination method to future work.

Author Contributions: Conceptualization, L.Y. and P.C.; methodology, L.Y.; software, P.C., Y.G. and Z.L.; validation, P.C., Y.G. and Z.L.; formal analysis, Z.L.; investigation, P.C., L.Y. and Z.L.; resources, L.Y.; data curation, P.C., Y.G. and Z.L.; writing—original draft preparation, L.Y. and P.C.; writing—review and editing, L.Y. and Z.L.; visualization, Y.G. and Z.L.; supervision, L.Y.; project administration, L.Y.; funding acquisition, L.Y. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China Grant Number [61773406] and the Natural Science Foundation of Hunan Grant Number [2021JJ30877]. And The APC was funded by [2021JJ30877].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: These tested datasets originate from UCI.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Taher, D.I.; Abu-Gdairi, R.; El-Bably, M.K.; El-Gayar, M.A. Decision-making in diagnosing heart failure problems using basic rough sets. *AIMS Math.* **2024**, *9*, 21816–21847. [CrossRef]
2. Zhang, Q.; Xie, Q.; Wang, G. A survey on rough set theory and its applications. *CAAI Trans. Intell. Technol.* **2016**, *1*, 323–333. [CrossRef]
3. Yin, L.; Cao, K.; Jiang, Z.; Li, Z. RA-MRS: A high efficient attribute reduction algorithm in big data. *J. King Saud Univ.-Comput. Inf. Sci.* **2024**, *36*, 102064. [CrossRef]
4. Qin, L.; Wang, X.; Yin, L.; Jiang, Z. A distributed evolutionary based instance selection algorithm for big data using Apache Spark. *Appl. Soft Comput.* **2024**, *159*, 111638. [CrossRef]
5. Akram, M.; Ali, G.; Alcantud, J.C.R. Attributes reduction algorithms for m-polar fuzzy relation decision systems. *Int. J. Approx. Reason.* **2022**, *140*, 232–254. [CrossRef]
6. Liu, G.; Feng, Y. Knowledge granularity reduction for decision tables. *Int. J. Mach. Learn. Cyber.* **2022**, *13*, 569–577. [CrossRef]
7. Yin, L.; Qin, L.; Jiang, Z.; Xu, X. A fast parallel attribute reduction algorithm using Apache Spark. *Knowl.-Based Syst.* **2021**, *212*, 106582. [CrossRef]
8. Xie, X.; Gu, X.; Li, Y.; Ji, Z. K-size partial reduct: Positive region optimization for attribute reduction. *Knowl.-Based Syst.* **2021**, *228*, 107253. [CrossRef]
9. Turaga, V.K.H.; Chebrolu, S. Rapid and optimized parallel attribute reduction based on neighborhood rough sets and MapReduce. *Expert Syst. Appl.* **2025**, *260*, 125323. [CrossRef]
10. Kumar, A.; Prasad, P.S.V.S.S. Enhancing the scalability of fuzzy rough set approximate reduct computation through fuzzy min–max neural network and crisp discernibility relation formulation. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104697. [CrossRef]
11. Sowkunta, P.; Prasad, P.S.V.S.S. MapReduce based parallel fuzzy-rough attribute reduction using discernibility matrix. *Appl. Intell.* **2022**, *52*, 154–173. [CrossRef]
12. Xu, W.; Yuan, K.; Li, W.; Ding, W. An Emerging Fuzzy Feature Selection Method Using Composite Entropy-Based Uncertainty Measure and Data Distribution. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 76–88. [CrossRef]
13. Sang, B.; Chen, H.; Yang, L.; Li, T.; Xu, W. Incremental Feature Selection Using a Conditional Entropy Based on Fuzzy Dominance Neighborhood Rough Sets. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 1683–1697. [CrossRef]
14. Ji, X.; Li, J.; Yao, S.; Zhao, P. Attribute reduction based on fusion information entropy. *Int. J. Approx. Reason.* **2023**, *160*, 108949. [CrossRef]
15. Fontes, D.O.L.; Vasconcelos, L.G.S.; Brito, R.P. Blast furnace hot metal temperature and silicon content prediction using soft sensor based on fuzzy C-means and exogenous nonlinear autoregressive models. *Comput. Chem. Eng.* **2020**, *141*, 107028. [CrossRef]
16. Nistala, S.H.; Kumar, R.; Parihar, M.S.; Runkana, V. metafur: Digital Twin System of a Blast Furnace. *Trans. Indian. Inst. Met.* **2024**, *77*, 4383–4393. [CrossRef]
17. Jiang, K.; Jiang, Z.; Xie, Y.; Pan, D.; Gui, W. Prediction of Multiple Molten Iron Quality Indices in the Blast Furnace Ironmaking Process Based on Attention-Wise Deep Transfer Network. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 2512114. [CrossRef]
18. Liu, C.; Tan, J.; Li, J.; Li, Y.; Wang, H. Temporal Hypergraph Attention Network for Silicon Content Prediction in Blast Furnace. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 2521413. [CrossRef]
19. Li, J.; Yang, C.; Li, Y.; Xie, S. A Context-Aware Enhanced GRU Network with Feature-Temporal Attention for Prediction of Silicon Content in Hot Metal. *IEEE Trans. Ind. Inf.* **2022**, *18*, 6631–6641. [CrossRef]

20. Wang, W.; Huang, B.; Wang, T. Optimal scale selection based on multi-scale single-valued neutrosophic decision-theoretic rough set with cost-sensitivity. *Int. J. Approx. Reason.* **2023**, *155*, 132–144. [CrossRef]
21. Shu, W.; Xia, Q.; Qian, W. Neighborhood multigranulation rough sets for cost-sensitive feature selection on hybrid data. *Neurocomputing* **2023**, *565*, 126990. [CrossRef]
22. Yang, J.; Kuang, J.; Liu, Q.; Liu, Y. Cost-Sensitive Multigranulation Approximation in Decision-Making Applications. *Electronics* **2022**, *11*, 3801. [CrossRef]
23. Su, H.; Chen, J.; Lin, Y. A four-stage branch local search algorithm for minimal test cost attribute reduction based on the set covering. *Appl. Soft Comput.* **2024**, *153*, 111303. [CrossRef]