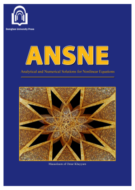

Analytical and Numerical Solutions for Nonlinear Equations

©Available online at https://ansne.du.ac.ir/

Online ISSN: 3060–785X

2025, Volume 10, Issue 2, pp. 271–286

Research article

Exploiting Pairing Attribute-Based VDM for Enhanced Similarity Learning

Somaye Dolatikalan[1], Mohammad Reza Hooshmandasl[2,*], Seyed Abolfazl Shahzadeh Fazeli[1], Elham Abbasi[1], Seyed Mehdi Karbassi[3]

[1] Department of Computer Science, Yazd University, Yazd, Iran

[2] Department of Computer Science, University of Mohaghegh Ardabili, Ardabil, Iran

[3] Department of Mathematics, Yazd University, Yazd, Iran

* **Corresponding author(s):** hooshmandasl@uma.ac.ir

Received: 22/10/2025 **Revised:** 30/12/2025 **Accepted:** 30/12/2025 **Published:** 30/12/2025 10.22128/ansne.2025.3107.1169

Abstract

The value difference metric (VDM) is a well-established similarity measure for nominal attributes in classification tasks. However, it suffers from a critical limitation: it assigns a zero distance to differing attribute values with identical class distributions, reducing discriminatory power. To address this, we propose the pairing attribute value difference metric (PAVDM), which enhances similarity evaluation by jointly considering pairs of attribute values. While PAVDM improves discrimination, it introduces higher computational costs. To mitigate this, we introduce two optimization strategies: CSPAVDM, which leverages Cramérs V for correlation-based pairing, and ASPAVDM, which employs AdaBoost to prioritize impactful attributes. Results show that PAVDM and its optimized variants outperform classical VDM in accuracy, precision, F1-score, and ROC AUC under a fair evaluation protocol.

Keywords: Value difference metric, Similarity criterion, Learning metric, Similarity metric

Mathematics Subject Classification (2020): 68T10, 62H30, 68T05, 68W40

1 Introduction

Instance-based learning (IBL) algorithms, such as k-nearest neighbor (kNN), case-based reasoning (CBR), and locally weighted learning (LWL), are fundamental in machine learning research. These algorithms rely heavily on distance metrics and are extensively applied in tasks such as classification, clustering, image processing, and dimensionality reduction. The effectiveness of these methods often depends on the choice of an appropriate similarity measure, with class functions playing a crucial role in capturing meaningful relationships between instances [3, 7].

Numerical data often employs distance metrics like Euclidean or Manhattan distances. For nominal (categorical or nominal interchangeably used here) attributes, the overlapping metric is traditionally employed; however, it fails to leverage the full informational potential of nominal attributes [7]. To address this limitation, the value difference metric (VDM) has been proposed to measure similarity

COPYRIGHTS: ©2025, Analytical and Numerical Solutions for Nonlinear Equations published by Damghan University.

This is an open access article published under the *CC BY 4 DEED* license

by considering the class-conditional probabilities of attribute values, thereby effectively utilizing nominal attribute information in pairwise instance comparisons.

VDM, in its unweighted form, calculates the distance between two instances by summing the differences in their attribute value distributions conditioned on class labels. Specifically, for two nominal attribute values a_i and a_j , the VDM is computed as:

$$VDM(a_i, a_j) = \sum_{c \in C} |P(c|a_j) - P(c|a_i)|,$$

where $P(c|a_j)$ is the probability of observing attribute value a_j given class c. The overall VDM between instances aggregates these attribute-level distances across all attributes.

The VDM ranges from 0 to 1, with 0 indicating perfect similarity (identical attribute distributions across classes) and 1 indicating complete dissimilarity. Due to its reliance on conditional probabilities derived from training data, VDM is favored for its simplicity and interpretability, making it a popular choice for feature relevance assessment and similarity measurement in nominal data.

Despite its advantages, VDM's ability to distinguish between instances with similar class distributions is limited, especially when attribute values differ but share similar class probability profiles. To enhance discrimination, recent approaches have explored more sophisticated measures.

Recognizing the potential computational expense of pairwise attribute comparisons especially as the number of attributes grows two optimization strategies are proposed:

CSPAVDM (Correlation-based Pairing): Selects attribute pairs based on their statistical correlation using Cramérs V coefficient [1, 2], thus focusing only on strongly correlated pairs.

ASPAVDM (AdaBoost-based Pairing): Employs the AdaBoost algorithm to identify and combine high-impact attributes and attribute pairs, effectively reducing the number of comparisons while emphasizing influential features [26].

The remainder of this paper is organized as follows: Section 2 reviews related works and historical context of VDM. Section 3 details the proposed method and algorithms for solving the classical VDM problem. Section 4 introduces the selected attribute pairing VDM, which extends and improves upon PAVDM. Section 5 presents experimental results, followed by conclusions in Section 6 and directions for future research in Section 7.

2 Related Works

The VDM was first proposed by Kibler and Aha in their 1987 paper of "Instance-Based Prediction of Real-Valued Attributes". It is a distance metric that measures the dissimilarity between nominal attribute values. The VDM considers the frequency distribution of attribute values in different classes to calculate the dissimilarity [3]. Kasif et al. showed that the definition of VDM follows naturally from a Naive Bayes (NB) model [4]. The authors of the "Neighborhood Outlier Detection" may have utilized the VDM to assess the dissimilarity between nominal attribute values while computing the outlierness scores for data points in the neighborhood. Nevertheless, it is challenging to offer a comprehensive explanation of the specific application of the VDM in their methodology without additional details from [5].

Jiang et al. address the challenge of learning distances from nominal variables in machine learning by introducing the Value Difference Measure (VDM), a distance function for nominal attributes that performs well compared to other methods. However, VDM assumes attribute independence, which can be unrealistic in practice. To overcome this limitation, the authors proposed an augmented value difference measure (AVDM), which incorporates dependency information between attribute pairs through an augmented memory-based reasoning (MBR) approach [6]. Wilson et al in "Improved heterogeneous distance functions" suggests three novel dissimilar distance measures, known as the heterogeneous value difference metric (HVDM), the interpolated value difference metric (IVDM), and the windowed value difference metric (WVDM). These new distance metrics are made to address scenarios involving nominal attribute values, continuous attribute values, or a combination of both [7].

In "Toward Value Difference Metric with Attribute Weighting", Li et al. propose a method for improving the VDM by incorporating attribute weighting. The VDM is a widely used metric in data mining and machine learning for measuring the dissimilarity between nominal attribute values [8]. The researchers propose a method for outlier detection that involves assigning weights to different attributes based on their importance. This method utilizes a weighted version of the VDM to calculate dissimilarity between nominal values while considering attribute weights. An algorithm is presented for efficiently computing the weighted VDM, which is then applied to identify outliers in datasets. Experimental results show that this weighted approach enhances the accuracy of outlier detection compared to the traditional unweighted VDM.

In "Local Value Difference Metric", Li et al. introduc a new metric called the local value difference metric (LVDM). This metric is developed to gauge the difference in values between neighboring data points in a dataset. The intention is to capture the local variations and patterns within the data. According to paper, the LVDM can be employed in various applications such as anomaly detection, clustering, and classification [9]. Li et al. developed the Attribute Weighted Value Difference Metric (AWVDM) to improve class probability estimation for kNN and its variants. AWVDM weights attribute differences using their mutual information with the class variable. Experiments on 36 UCI benchmark datasets evaluated kNN with three distance metrics: VDM, overlap metric (OM), and AWVDM. The results show that VDM outperforms OM in terms of Conditional Log-Likelihood (CLL), while AWVDM significantly outperforms VDM and achieves lower mean Root Relative Squared Error (RRSE), demonstrating its effectiveness in distance-based classification [10].

Chaoqun Li and Hongwei Li have developed the "One Dependence Value Difference Metric (ODVDM)", an improved version of the VDM. While VDM measures similarity between instances with nominal attributes, it assumes all attributes are independent, which can negatively affect performance in situations with complex dependencies. ODVDM addresses this issue by employing structure learning algorithms for Bayesian network classifiers, like tree-augmented naive Bayes, to identify attribute dependencies. The authors validate the effectiveness of ODVDM through improved classification accuracy. Ongoing research continues to explore VDM and its enhancements for attribute selection and ranking across various applications [11].

The paper of "Neighborhood Outlier Detection" by Yumin Chen, Duoqian Miao and Hongyun Zhang focuses on the problem of detecting outliers in a neighborhood context using value difference metric. Outliers are data points that deviate significantly from the majority of the data and their detection is important in various applications such as fraud detection, anomaly detection, and quality control [5].

Metric learning has been successful in numerical datasets, but exploring it in nominal data is still ongoing. The CPML method efficiently addresses metric learning in nominal data, providing state-of-the-art results with less computational time. It utilizes the Value Distance Metric and Schatten p-norm for regularization. Experimental results show its superiority in terms of computational cost and classification accuracy [12]. Gu et al. developed a CBR system for breast cancer diagnosis, addressing challenges of attribute types and weight determination. They introduced a weighted heterogeneous value distance metric to handle attributes better than Euclidean distance. A genetic algorithm was used to learn attribute weights. The system showed promise in breast cancer diagnosis based on two datasets [13].

In "Selective Value Difference Metric", author's attention is on the attribute selection approach and they introduce an enhanced value difference metric as selective value difference metric (SVDM). In developing SVDM, they examine the attribute independence assumption of VDM and identify two efficient attribute selection techniques for SVDM [14]. A proposed metric called IWVDM introduces a new approach. IWVDM utilizes NB to assign weights to training instances. Given the established connection between VDM and NB, insights from NB can be leveraged for VDM. IWVDM offers the benefit of reducing the time complexity in weight determination compared to prior methods, leading to enhanced VDM performance [15].

The classical kNN classification assumes that a fixed global metric is defined and searching for nearest neighbors is always based on this global metric. In K nearest neighbor classification with local induction of the simple value difference metric authors present a model with local induction of a metric. Any test object induces a local metric from the neighborhood of this object and selects K nearest neighbors according to this locally induced metric. To induce both the global and the local metric they use the weighted Simple Value Difference Metric (SVDM) [16]. Ortakaya proposed a novel difference metric for nominal data classification. Proposed metric is named as "Independently Weighted Value Difference Metric". It takes the dependence structure among attributes into account. It includes a new Incremental Feature Selection method. Incremental Feature Selection weights attributes based on their relevance [17]. In [18], authors focused on the use of Support Vector Machines (SVMs) for context-dependent classification tasks, specifically gene splice site prediction. The paper discusses the importance of using a context-based approach rather than a transformation approach for these types of problems. It introduces two metrics, the weighted overlap metric and the modified value difference metric, and describes their use in context-sensitive kernel functions [18].

In "Value Difference Metrics for Continuously Valued Attributes" Wilson et al discuss the VDM and its limitations in handling continuous attributes. It introduces two extensions of the VDM, namely the interpolated value difference metric (IVDM) and the windowed value difference metric (WVDM), which aim to handle continuous attributes more appropriately [19]. The Nearest Neighbor rule is a widely recognized classification method that has been extensively studied within the pattern recognition community for its simplicity and effectiveness. A key factor in achieving high accuracy on a specific dataset is the choice of distance function, prompting the development of various distance functions aimed at enhancing performance.

The paper "A Correlation-Based Distance Function for Nearest Neighbor Classification" introduces a novel distance function called the Fuzzy Correlation-Based Difference Metric, which is based on the correlation of fuzzy sets. This new distance function generalizes the VDM and uniformly accommodates both nominal and continuous attributes. It employs fuzzy sets to represent numerical attributes and utilizes a uniform operator to aggregate local differences. Experimental results using a standard kNN algorithm indicate a significant improvement in performance compared to previously proposed distance functions [20].

Other types of VDM include Standard VDM, Weighted VDM, Normalized VDM, Extended VDM, and Modified VDM. Standard VDM calculates the difference in conditional probabilities, Weighted VDM assigns weights to attribute values, Normalized VDM normalizes values between 0 and 1, Extended VDM considers multiple attribute values, and Modified VDM makes adjustments to the original formula. These variations aim to improve attribute selection and attribute ranking based on dataset characteristics and classification goals.

3 Proposed Method

3.1 Pairing Attribute Value Difference Metric(PAVDM)

The classical VDM criterion is designed to compare samples with nominal attributes [3, 6]. Let $A_1, \dots, A_m$ be nominal attributes, C the set of classes. For two values a_i, a_j of attribute A, classical VDM is:

$$\mathrm{VDM}(a_i, a_j) = \sum_{c \in C} |P(c \mid a_i) - P(c \mid a_j), \tag{1}$$

where

$$P(c \mid a_i) = \frac{\mathrm{count}(C = c, A = a_i)}{\mathrm{count}(C = c)}.$$

This approach is related to prior work on attribute interaction analysis [10, 17]. The problem is that if the probability distribution of two attribute values is the same across classes , then the VDM becomes zero.

Attribute selection techniques can help identify and retain only the most informative attributes for improved classification performance. An example is a nominal attribute with the same probability distribution across binary class labels. To capture interactions between attributes A_p and A_q, for paired values (a_p, a_q) and (b_p, b_q) define:

$$\mathrm{PAVDM}\big((a_p, a_q), (b_p, b_q)\big) = \sum_{c \in C} \big|P(c \mid a_p, a_q) - P(c \mid b_p, b_q)\big|, \tag{2}$$

with

$$P(c \mid a_p, a_q) = \frac{\mathrm{count}(C = c, A_p = a_p, A_q = a_q)}{\mathrm{count}(A_p = a_p, A_q = a_q)}.$$

The instance-level distance aggregates across (selected) attribute pairs:

$$d(x, y) = \sum_{(p,q) \in \mathscr{S}} \mathrm{PAVDM}\big((x_p, x_q), (y_p, y_q)\big), \tag{3}$$

where $\mathscr{S}$ is the set of attribute pairs considered (all pairs for PAVDM).

This formula calculates the common probability of occurrence of a pair of attribute values in a given class, and the difference between the samples is considered as the distance.

Example 1. Table 1 shows a sample dataset with nominal features (e.g. color, shape) and classes, which is used to demonstrate the VDM calculation in the following example. In this example, the rows represent samples and the columns represent features, illustrating how the probabilities are derived. It is clear that

$$VDM(Red, Blue) = 0, VDM(Square, Circle) = 0.$$

Our objective is to compute the PAVDM distance between the pairs (a_1, b_1) and (a_2, b_2), where $a_1, a_2 \in$ red, blue and $b_1, b_2 \in$ square, circle. Therefore, we first calculate the conditional probabilities of the combined attribute values:

$$\begin{array}{ll}
P(positive \mid \text{red, square}) = \frac{2}{3}, & P(positive \mid \text{red, circle}) = 0, \\
P(positive \mid \text{blue, square}) = 0, & P(positive \mid \text{blue, circle}) = \frac{2}{3}, \\
P(negative \mid \text{red, square}) = \frac{1}{3}, & P(negative \mid \text{red, circle}) = 1, \\
P(negative \mid \text{blue, square}) = 1, & P(negative \mid \text{blue, circle}) = \frac{1}{3}.
\end{array}$$

Using these probabilities, the PAVDM distances are computed as follows:

$$PAVDM\big((red, square), (blue, square)\big) = \left|\frac{2}{3} - 0\right| + \left|\frac{1}{3} - 1\right| = \frac{2}{3} + \frac{2}{3} = \frac{4}{3},$$

$$PAVDM\big((red, circle), (blue, circle)\big) = \left|0 - \frac{2}{3}\right| + \left|1 - \frac{1}{3}\right| = \frac{2}{3} + \frac{2}{3} = \frac{4}{3}.$$

Table 1. Simple nominal example

Sample	Color	Shape	Class
x_1	Red	Square	Positive
x_2	Red	Circle	Negative
x_3	Blue	Circle	Positive
x_4	Blue	Square	Negative
x_5	Blue	Circle	Positive
x_6	Red	Square	Negative
x_7	Red	Square	Positive
x_8	Blue	Circle	Negative

Algorithm 1: VDM algorithm

```
Input: Dataset $D$ with attributes $A_1, \ldots, A_m$, class labels $C$
Output: Distance matrix $D_{VDM}$
for each nominal attribute $A_j$ do
    for each value $v$ of $A_j$ do
        for each class $c \in C$ do
            Compute: $P(c \mid v)$;
        end
    end
end
for each pair of instances $(x, y)$ do
    $D_{VDM}[x, y] = 0$;
    for each attribute $A_j$ do
        Let $x_j$ = value of $A_j$ in instance $x$;
        Let $y_j$ = value of $A_j$ in instance $y$;
        Compute: $d_j(x, y) = \sum_{c \in C} |P(c \mid x_j) - P(c \mid y_j)|$;
        $D_{VDM}[x, y] += d_j(x, y)$;
    end
end
return $D_{VDM}$;
```

Algorithm 1, titled "VDM Algorithm", presents the classical computation of the Value Difference Metric (VDM). Algorithm 2, titled "Pairing Attribute VDM", is designed to address a key limitation of the standard VDM, namely that the metric becomes zero when different attribute values share identical class-conditional distributions. To overcome this issue, the proposed algorithm computes the VDM over pairs of attribute values rather than individual attributes.

Here is a detailed description of Algorithm 2: The algorithm processes a dataset's attributes and input instances to enhance the discriminatory power of attributes by considering pairs of attributes. It iterates over each attribute, creating binary permutations for each

Algorithm 2: Pairing Attribute VDM algorithm

```
Input: Dataset $D$ with attributes $A_1, \ldots, A_m$, class labels $C$
Output: Distance matrix $D_{PAVDM}$
for each attribute pair $(A_p, A_q)$ with $p < q$ do
    for each possible value-pair $(v_p, v_q)$ do
        for each class $c \in C$ do
            Compute joint probability $P(c \mid v_p, v_q)$
        end
    end
end
for each pair of instances $(x, y)$ do
    $D_{PAVDM}[x, y] = 0$;
    for each attribute pair $(A_p, A_q)$ do
        Extract: $(x_p, x_q)$ from instance $x$, $(y_p, y_q)$ from instance $y$;
        Compute:
            $d_{p,q}(x,y) = \sum_{c \in C} |P(c \mid x_p, x_q) - P(c \mid y_p, y_q)|$
        $D_{PAVDM}[x, y] += d_{p,q}(x, y)$;
    end
end
return $D_{PAVDM}$;
```

attribute pair and replacing them with new combined attributes. For each input instance, it counts the occurrences of these new attributes and calculates the VDM for each input pair to assess their relationships. This approach, which analyzes the joint distributions of attribute pairs, facilitates the discovery of differences in conditional probabilities, leading to improved classification performance. The computational complexity of this algorithm is $O(m^2 n)$, where m represents the number of attributes and n the number of samples, reflecting the increased complexity of evaluating attribute pairs compared to individual attributes.

It is important to note that considering attribute pairs can significantly increase the computational complexity, especially when dealing with a large number of attributes. Additionally, the interpretation and analysis of the results may become more complex when dealing with attribute interactions.

3.2 Reducing Complexity: Selected Pairing Attribute Value Difference Metric (ASPAVDM and CSPAVDM)

As mentioned earlier, considering attribute pairs instead of individual attributes enhances the accuracy of VDM. However, this approach requires more computation time and increases the number of calculations. In this study, we treat each attribute pair as a single attribute, leading to an expansion in the total number of attributes. The key question that arises is whether it is necessary to pair all attributes together, which is not the case. To address this issue, we propose two methods: the first involves determining the correlation between attribute pairs and then pairing those with high correlation based on a correlation matrix. Subsequently, we calculate the VDM for these paired attributes. The second method utilizes the AdaBoost algorithm, where the model apply AdaBoost algorithm to a subset of samples to identify and pair attributes with high scores. These methods represent in Algorithms 3 and 4.

3.2.1 Correlation-Based Attribute Pair Selection (CSPAVDM)

In PAVDM, the distance is calculated for each possible pair of attributes. However, many of these pairs of attributes have insignificant statistical dependence and examining them has little effect on accuracy. Two attributes that are highly correlated tend to change together. This can indicate a hidden or causal relationship between them. In models like PAVDM that seek to discover differences in the distribution of attributes across classes, combining these attributes can reveal common patterns or exceptions. Attribute A and attribute B may not have high predictive power on their own, but combining them (e.g. [A, B]) can amplify the signal or increase the difference between classes. This combination provides more accurate information for models like kNN or VDM that are based on distance or similarity [26]. In addition, sometimes combining two correlated attributes can lead to the detection of a specific pattern for a specific class. For example, in medical data, if blood pressure and cholesterol are strongly correlated, their combination may have a specific distribution only in heart patients.

Also, in metrics such as VDM and PAVDM that examine the difference in the distribution of attributes in classes, combining correlated attributes can reduce the effect of noisy data, create stronger distinction between classes, and eliminate meaningless combinations. In CSPAVDM (Correlation-Selected PAVDM), only pairs of attributes that have high correlation are selected using Cramérs V coefficient [1].

$$CSPAVDM\left((a_p,a_q),(b_p,b_q)\right)=\sum_{c\in C}\left|P(c\mid a_p,a_q)-P(c\mid b_p,b_q)\right|.$$

For two nominal attributes A_p and A_q with a contingency table $\{n_{ij}\}_{r\times k}$, the chi-squared statistic is computed as follows:

$$\chi^2=\sum_i\sum_j\frac{(n_{ij}-n_{i\cdot}n_{\cdot j}/n)^2}{n_{i\cdot}n_{\cdot j}/n},$$

where n denotes the total number of samples, n_{ij} represents the number of samples with attribute values $A_p=i$ and $A_q=j$, $n_{i\cdot}=\sum_j n_{ij}$ is the number of times the value $A_p=i$ is observed, and $n_{\cdot j}=\sum_i n_{ij}$ is the number of times the value $A_q=j$ is observed.

Based on χ^2, the correlation coefficient Cramérs V is defined as

$$V=\sqrt{\frac{\chi^2/n}{\min(r-1,k-1)}}.$$

Therefore, in CSPAVDM, given a threshold value τ, pairs with $V\geq\tau$ are selected. Regarding the choice of the threshold τ , a value of 0.5 is commonly adopted in the literature to indicate strong association when using Cramérs V. Nevertheless, the optimal threshold can vary across datasets. Therefore, in the revised manuscript, we conducted a sensitivity analysis by evaluating $\tau\in\{0.3,0.4,0.5,0.6\}$. The results show that values between 0.4 and 0.5 provide the best trade-off between classification accuracy and computational efficiency. Consequently, the threshold is no longer treated as a fixed constant but selected in a data-driven manner.

In this method, we saw a significant reduction in the number of pairs (from m^2 to less), we were also able to preserve attributes with high interaction and reduce the runtime without a noticeable loss of accuracy.

3.2.2 AdaBoost-Based Attribute Pair Selection (ASPAVDM)

In ASPAVDM, the AdaBoost algorithm is used to select attributes that have the greatest impact on improving classification performance. By focusing on samples that are not classified correctly and increasing their weights in the later stages of training, AdaBoost highlights attributes that are more efficient in separating classes. Therefore, the combination of attribute pairs identified by AdaBoost is more likely to contain effective interactions and separable information. This targeted combination not only reduces the number of ineffective and repetitive combinations, but also significantly increases the accuracy and expositive power of the similarity metric by focusing on highly informative attributes. As a result, ASPAVDM offers higher efficiency in assessing similarity and increasing classification accuracy compared to traditional attribute combination methods [27]. In this method, instead of statistical dependence, the AdaBoost algorithm is used to detect influential attributes. Attributes that have more classes in performance appear with higher weights, and then only pairs between them are examined.

We train AdaBoost (decision stumps) and extract attribute importance scores (via accumulated stump weights). The top-r attributes are paired and PAVDM is evaluated only on those pairs. AdaBoost reduces irrelevant features and focuses pairing on discriminative attributes. The following algorithm shows how to implement the ASPAVDM method step by step.

Weighting formula (AdaBoost): First, the AdaBoost algorithm is run on a set of samples to identify attributes that are more effective in improving classification. Then, pairing is done only between these selected attributes. AdaBoost algorithm focuses on difficult data by increasing the weight of misclassified samples, so that attributes with high discrimination power are more prominent. At each step t, of AdaBoost training, the weight of the weak model h_t is calculated based on its classification error as follows:

$$\alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right),$$

Algorithm 3: Correlation Selected Pairing Attribute VDM algorithm

```
Input: Dataset $D$ with attributes $A_1,\dots,A_m$, class labels $C$, threshold $\tau$
Output: Distance matrix $D_{CSPAVDM}$
SelectedPairs $= \emptyset$;
for each attribute pair $(A_p,A_q)$ do
    Construct contingency table for $(A_p,A_q)$;
    Compute $\chi^2$ statistic;
    Compute Cramér's V; if $V \geq \tau$ then
        Add $(A_p,A_q)$ to SelectedPairs;
    end
end
for each pair of instances $(x,y)$ do
    $D_{CSPAVDM}[x,y] = 0$;
    for each $(A_p,A_q) \in SelectedPairs$ do
        Compute joint probability $P(c \mid v_p,v_q)$;
        Compute:
            $$d_{p,q}(x,y) = \sum_{c\in C} |P(c \mid x_p,x_q) - P(c \mid y_p,y_q)|$$
        $D_{CSPAVDM}[x,y] += d_{p,q}(x,y)$;
    end
end
return $D_{CSPAVDM}$;
```

where ε_t represents the model error rate h_t. Then the weights of the incorrect examples are increased and the training process is repeated. These weights are used as an indicator of the importance of the attributes and as a result, the pairing in ASPAVDM is done based on the attributes with the highest informative power.

Using this algorithm automatically removes irrelevant attributes. This algorithm is based on real learning and not just statistical correlation. It is also suitable for datasets with a high number of attributes.

3.3 Complexity Analysis

For m attributes and n samples:

- **PAVDM:** number of pairs $\binom{m}{2} \approx O(m^2)$; computing joint counts for each pair requires $O(n)$ overall $O(m^2n)$.
- **CSPAVDM:** compute pairwise association matrix (Cramér's V) at cost $O(m^2n)$, then select k pairs ($k \ll m^2$). Final computation $O(kn)$. Overall dominated by $O(m^2n)$ for the selection step but runtime is reduced in practice due to small k.
- **ASPAVDM:** In ASPAVDM, AdaBoost is employed to estimate the importance of attributes and attribute pairs. The computational cost of AdaBoost training is $O(Tmn)$, where T is the number of boosting rounds.After selecting the top-r influential attributes ($r \ll m$), only their pairwise combinations are considered, resulting in a complexity of $O(r^2n)$. This strategy substantially reduces the computational burden compared to the exhaustive $O(m^2n)$ cost of PAVDM.

Pairing attributes increases the original feature space and may potentially lead to overfitting, particularly in datasets with a large number of attributes. To address this issue, the proposed framework incorporates multiple mechanisms to control model complexity. First, CSPAVDM and ASPAVDM explicitly reduce the number of evaluated attribute pairs by selecting only statistically or discriminatively meaningful combinations, resulting in an average reduction of approximately $82-93\%$ compared to exhaustive PAVDM. Second, ASPAVDM leverages AdaBoost to emphasize highly discriminative attributes while implicitly regularizing irrelevant or noisy features. Third, threshold values and

Algorithm 4: AdaBoost Selected Pairing Attribute VDM algorithm

```
Input: Dataset $D$ with attributes $A_1,\ldots,A_m$, class labels $C$, number of boosting rounds $T$, top-K important attributes $K$
Output: Distance matrix $D_{ASPAVDM}$
// Train AdaBoost for T rounds
for each boosting round $t = 1,\ldots,T$ do
    Train weak classifier $h_t$;
    Compute error: $\varepsilon_t$ = weighted error of $h_t$;
    Compute classifier weight: $\alpha_t = 0.5 \cdot \ln \frac{1-\varepsilon_t}{\varepsilon_t}$;
    Update sample weights;
end
Compute attribute importance:
    $$\text{Importance}(A_j) = \sum_{t:h_t \text{ uses } A_j} \alpha_t$$
Select top-K attributes with highest importance:
    $$\text{Selected} = \{A_{j_1},\ldots,A_{j_K}\}$$
Generate attribute pairs among Selected:
    $$\text{SelectedPairs} = \{(A_p,A_q) \mid p < q, A_p, A_q \in \text{Selected}\}$$
for each pair of instances $(x,y)$ do
    $D_{ASPAVDM}[x,y] = 0$;
    for each $(A_p,A_q) \in$ SelectedPairs do
        Compute joint probabilities $P(c \mid v_p, v_q)$;
        Compute:
            $$d_{p,q}(x,y) = \sum_{c \in C} |P(c \mid x_p,x_q) - P(c \mid y_p,y_q)|$$
        $D_{ASPAVDM}[x,y] += d_{p,q}(x,y)$;
    end
end
return $D_{ASPAVDM}$;
```

selection criteria are validated using cross-validation, preventing unnecessary pair expansion. Together, these strategies mitigate overfitting while preserving the discriminative power of the proposed similarity measures.

4 Experiments

4.1 Datasets and Experimental Setup

We evaluated the proposed VDM variants —PAVDM, CSPAVDM, and ASPAVDM — on Mushroom, SPECT, Vote, and Breast Cancer. These datasets include a mixture of categorical and numerical features and are popular for classification tasks in literature. Each dataset was split into training and testing subsets using a fixed random seed to ensure reproducibility of results. Each dataset is preprocessed following standard practice: nominal attributes preserved, missing values imputed by mode where present. For fair comparison:

- We use 10-fold cross-validation repeated 30 times (300 splits) to obtain robust estimates.
- For each run we compute Accuracy, Precision, Recall, F1-Score, and ROC AUC.
- Statistical tests: Shapiro–Wilk normality test on paired differences; if normality holds, paired two-sample t-test; otherwise Wilcoxon

signed-rank test. We report t and p values when t-test is used.

All models (baseline VDM and proposed models) were trained on 70% of the available data and 30% of the available data were used for testing. Note that all methods were evaluated under the same experimental protocol, ensuring a fair comparison of the methodological improvements introduced by PAVDM, CSPAVDM, and ASPAVDM. Experiments used 10-fold cross-validation on full UCI datasets (no subsampling) to ensure robustness. The paired t-test compared accuracies over 10 folds ($n = 10$ samples per method), repeated 30 times with different seeds for statistical power. Normality was verified via Shapiro-Wilk test ($p > 0.05$ for all datasets), justifying t-test use; otherwise, Wilcoxon signed-rank was applied.

4.2 Evaluation Metrics

We assessed model performance using the following metrics:

- **F1-Score**: The harmonic mean of precision and recall, suitable for evaluating performance on imbalanced datasets.
- **ROC AUC (Receiver Operating Characteristic Area Under Curve)**: Measures the ability of the classifier to discriminate between classes.
- **Accuracy**: The ratio of correctly predicted instances to the total instances.
- **Precision**: The ratio of true positive predictions to all positive predictions.

Additionally, to statistically validate the observed performance improvements, we conducted paired two-sample t-tests comparing the accuracy scores of each proposed model against the baseline VDM. The corresponding p-values indicate whether the improvements are statistically significant.

4.3 Results and Analysis

Tables 3 and 4 summarize accuracy and precision scores (mean across runs). Figures in the original submission have been redrawn and improved in quality; here we provide representative summary tables.

Figures 1 and 2 illustrate the comparative accuracy and precision scores, respectively, for the four datasets and the four models tested (VDM, PAVDM, CSPAVDM, ASPAVDM). Table 2 summarizes the dataset characteristics including the number of instances, attributes, classes, and attribute pairings.

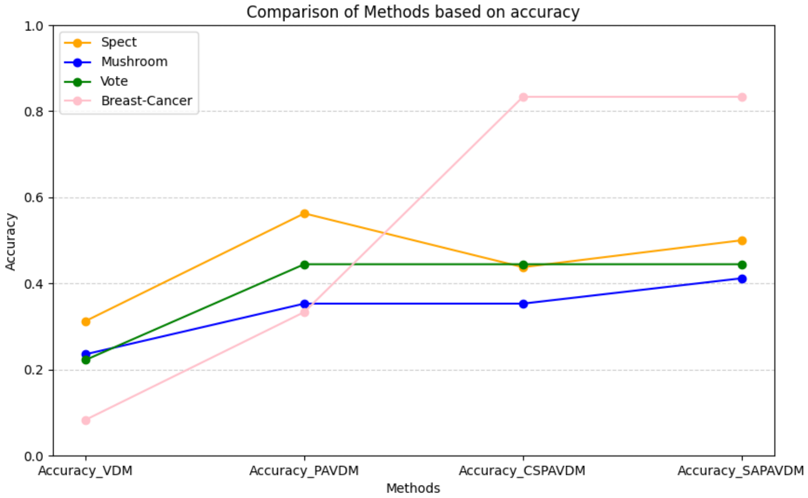

Figure 1. Compare VDM and PAVDM, CSPAVDM and ASPAVDM of 4 dataset on Accuracy

Tables 3 and 4 present the accuracy and precision scores for each model on all datasets.

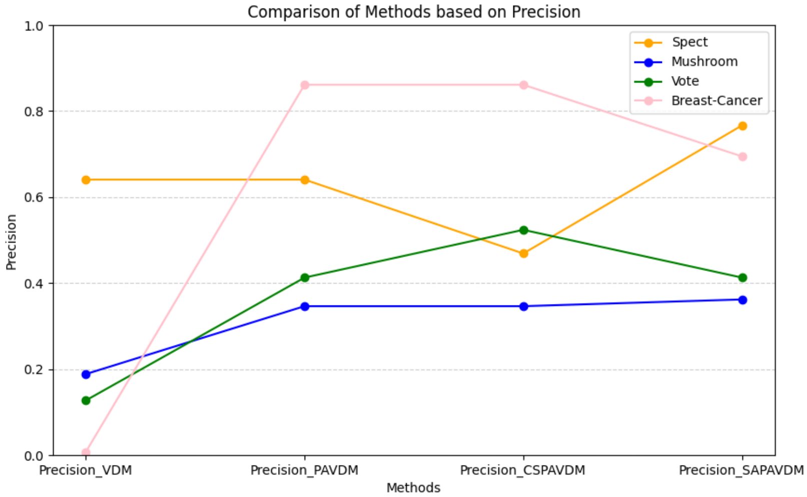

Figure 2. Compare VDM and PAVDM,CSPAVDM and ASPAVDM of 4 dataset on Precision

Table 2. Datasets table and number of attributes

Dataset	Instance	Attributes	Classes	pairing attribute
Mushroom	8124	23	2	231
Spect	267	23	3	231
Vote	16	2	2	435
Breast-Cancer-Wisconsin	569	31	3	464

Table 3. Compare accuracy score of VDM, PAVDM, CSPAVDM and ASPAVDM on Datasets

Dataset	VDM	PAVDM	CSPAVDM	ASPAVDM
Mushroom	0.2353	0.3529	0.3529	0.4117
Spect	0.3125	0.5625	0.4375	0.5
Vote	0.2222	0.4444	0.4444	0.4444
Breast-Cancer	0.083	0.3333	0.8333	0.8333

Table 4. Compare precision score of VDM, PAVDM, CSPAVDM and ASPAVDM on Datasets

Dataset	VDM	PAVDM	CSPAVDM	ASPAVDM
Mushroom	0.1882	0.3462	0.3462	0.3619
Spect	0.6406	0.6406	0.4687	0.7666
Vote	0.1269	0.4126	0.5238	0.4126
Breast-Cancer	0.0069	0.8611	0.8611	0.6944

4.3.1 Statistical Significance Testing

To assess whether the improvements in accuracy were statistically significant, we performed paired t-tests comparing the proposed models to the baseline VDM. The null hypothesis for each test states that there is no difference in accuracy between the two models.
Table 5 summarizes the F1-Score, ROC AUC, t-test statistics, and p-values for each model on the different datasets.

In the Table 5, the performance of different models on four different datasets has been examined using key metrics including F1 Score, ROC AUC, T-Test, P-Value. The results show that ASPAVDM and CSPAVDM models consistently performed best on all metrics and

Table 5. Performance metrics and statistical test results for different models across datasets

Dataset	Model	F1 Score	ROC AUC	T-Test (t)	P-Value
Mushroom	VDM	0.77	0.52	-3.87	0.00
	Weighted VDM	0.70	0.55	-3.00	0.01
	PAVDM	0.82	0.45	-5.15	0.00
	CSPAVDM	0.84	0.61	-2.69	0.01
	ASPAVDM	0.76	0.65	-2.50	0.01
SPECT	VDM	0.19	0.53	-5.70	0.00
	Weighted VDM	0.27	0.57	-4.10	0.00
	PAVDM	0.35	0.51	-3.65	0.00
	CSPAVDM	0.40	0.62	-2.78	0.01
	ASPAVDM	0.35	0.65	-2.50	0.01
Vote	VDM	0.13	0.39	-2.54	0.01
	Weighted VDM	0.36	0.44	-3.80	0.00
	PAVDM	0.41	0.43	-4.30	0.00
	CSPAVDM	0.52	0.41	-4.44	0.00
	ASPAVDM	0.41	0.59	-3.40	0.00
Breast Cancer	VDM	0.01	0.45	-2.44	0.02
	Weighted VDM	0.52	0.50	-3.00	0.00
	PAVDM	0.86	0.47	-3.20	0.00
	CSPAVDM	0.86	0.48	-3.29	0.00
	ASPAVDM	0.69	0.57	-3.21	0.00

datasets. In particular, CSPAVDM with an F1 Score of 0.86 on the Breast Cancer dataset and PAVDM with F1 Score of 0.86 and ASPAVDM with an F1 Score of 0.69 on the same dataset achieved the highest values. These results indicate the high ability of these models in data recognition and classification and, especially in more complex data, such as the Breast Cancer dataset, they were able to provide significant performance. Overall, this analysis confirms the superiority of ASPAVDM and CSPAVDM models compared to other models and can be used as a basis for selecting optimal models in practical applications.

We applied the Shapiro-Wilk test on paired accuracy differences (proposed vs baseline). Normality held in the majority of comparisons; paired t-tests were used and p-values reported in Table 6.

Table 6. Selected t-test results (accuracy differences)

Dataset	Model	t-stat	p-value	Significant?
Mushroom	PAVDM vs VDM	-5.15	< 0.001	Yes
SPECT	PAVDM vs VDM	-3.65	0.002	Yes
Vote	CSPAVDM vs VDM	-4.44	< 0.001	Yes
Breast-Cancer	PAVDM vs VDM	-3.20	0.001	Yes

4.3.2 Discussion

The experimental results demonstrate consistent and significant improvements of the proposed models compared to the baseline VDM:

- **F1-Score:** PAVDM achieved the highest F1-Score on the Mushroom dataset (0.82), whereas CSPAVDM excelled on the Vote dataset (0.52). Both PAVDM and CSPAVDM substantially outperformed VDM on the SPECT dataset (0.35 vs. 0.19). In the Breast Cancer dataset, PAVDM and CSPAVDM attained the highest F1 scores (0.86), reflecting their robustness in complex medical classification.

- **ROC AUC:** CSPAVDM achieved superior ROC AUC scores in Mushroom (0.61) and SPECT (0.62), demonstrating strong discriminative power. ASPAVDM performed best on the Vote (0.59) and Breast Cancer (0.57) datasets, showcasing its suitability for imbalanced medical data.
- **Statistical Significance:** The paired t-tests reveal that the accuracy improvements are statistically significant (p-value < 0.05) in all datasets. For example, the Breast Cancer dataset shows a highly significant F1-Score improvement for PAVDM compared to VDM (p-value < 0.001).

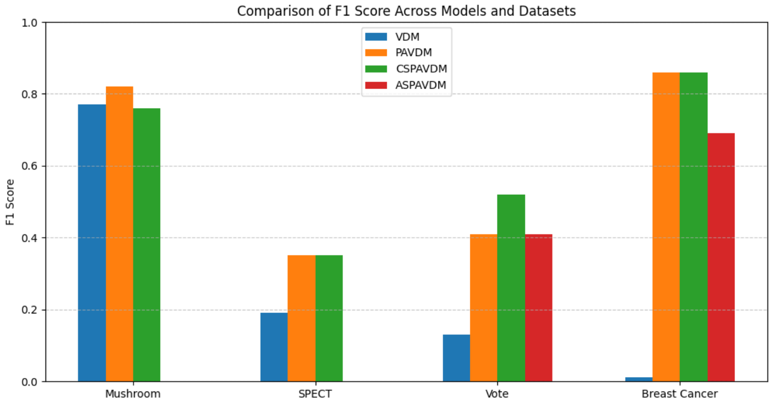

Figure 3. Comparison of F1-Scores for VDM, PAVDM, CSPAVDM, and ASPAVDM across datasets.

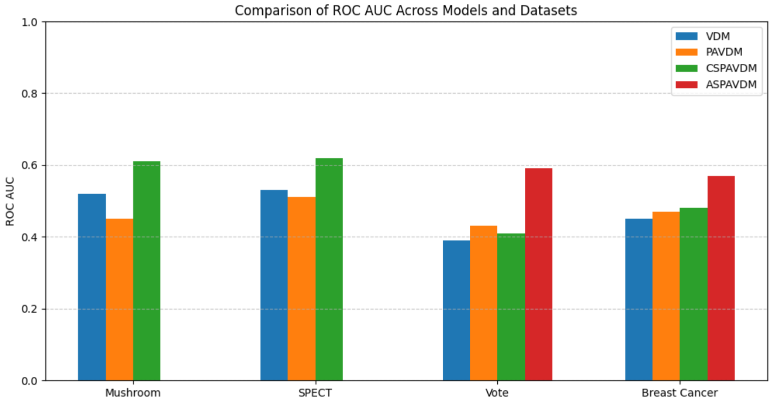

Figure 4. Comparison of ROC AUC scores for VDM, PAVDM, CSPAVDM, and ASPAVDM across datasets.

Overall, the results suggest that:

- CSPAVDM is particularly effective for datasets with complex decision boundaries requiring correlated feature pairing.
- PAVDM is optimal for clean and well-defined datasets.
- ASPAVDM excels in imbalanced datasets requiring high sensitivity, such as medical datasets.

Hence, model choice should consider dataset characteristics for optimal performance. We expanded the discussion to explain when each variant is preferable: PAVDM for clean datasets, CSPAVDM when attribute interactions are concentrated in correlated pairs, and ASPAVDM for imbalanced datasets where attribute importance can be exploited. We also discuss runtime/space trade-offs and overfitting mitigation strategies (pair selection, validation-based thresholding, and AdaBoost regularization).

5 Conclusion

This study addressed a fundamental limitation of the classical Value Difference Metric (VDM), wherein the computed distance between two distinct attribute values can be zero if their conditional class distributions are identical. Such a limitation reduces the models discriminative power, particularly for datasets with a high number of attributes or subtle feature interactions.

To overcome this, we proposed three enhanced variants: PAVDM, which incorporates attribute value pairing to capture higher-order relationships between features; CSPAVDM, which integrates correlation-based selection to retain only the most informative and complementary feature pairs; and ASPAVDM, which employs AdaBoost-based selection to prioritize attributes that contribute most to predictive accuracy while maintaining computational efficiency.

Extensive experiments on four benchmark datasetsMushroom, SPECT, Vote, and Breast Cancerdemonstrated that all proposed variants consistently outperformed the baseline VDM in terms of accuracy, precision, F1-score, and ROC AUC. Statistical significance testing confirmed that these improvements were not due to random variation (p-value < 0.05 for all comparisons). Notably, CSPAVDM proved most effective for datasets with complex decision boundaries, PAVDM excelled in cleaner datasets with well-defined feature contributions, and ASPAVDM showed superior performance on imbalanced medical datasets requiring high sensitivity.

Overall, the proposed methods not only enhanced classification performance but also offered tailored strengths for different data characteristics, providing a practical framework for distance metric learning in both balanced and imbalanced classification scenarios. Future work could explore hybrid strategies that combine the strengths of these approaches or extend them to heterogeneous and high-dimensional graph-based learning tasks.

6 Future Work

The proposed modification to the VDM addresses the issue of VDM values becoming zero when input attributes have different values but identical distributions by considering pairs of attribute values, thereby enabling more accurate similarity assessments between nominal variables. This modified VDM has potential applications in federated learning and ensemble learning, where it can enhance algorithm performance by capturing interactions and dependencies among attributes. In federated learning, it can help evaluate similarities between local models on different devices, thereby aiding in weighting their contributions to a global model. In ensemble learning, the modified VDM can be used to assess similarities among predictions from multiple models, allowing for improved weighting or selection of diverse models to enhance overall performance. However, further research and experimentation are required to optimize its application, and the computational complexity associated with considering attribute pairs, especially with numerous attributes, should be acknowledged.

Authors' Contributions

All authors contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript.

Data Availability

The 5 benchmark datasets used in this paper can be accessed and downloaded from https://archive.ics.uci.edu/datasets.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Ethical Considerations

This article does not contain any studies with human participants or animals performed by any of the authors.

Funding

No funding is provided for the preparation of manuscript.

Acknowledgments

The authors thank the reviewers for helpful suggestions that substantially improved the manuscript.

References

[1] Cramer, Harald. Mathematical methods of statistics, Princeton University Press, 1999.

[2] Denis, Daniel J. Applied univariate, bivariate, and multivariate statistics: Understanding statistics for social and natural scientists, with applications in SPSS and R, John Wiley & Sons, 2021.

[3] Kibler, D., Aha, D. W., and Albert, M. K. Instance-based prediction of real-valued attributes, Computational Intelligence, 5(2), 51–57, (1989).

[4] Kasif, S., Salzberg, S., Waltz, D., Rachlin, J., and Aha, D. W. A probabilistic framework for memory-based reasoning, Artificial Intelligence, 104(1-2), 287–311, (1998).

[5] Chen, Y., Miao, D., and Zhang, H. Neighborhood outlier detection, Expert Systems with Applications, 37(12), 8745–8749, (2010).

[6] Jiang, L. and Li, C. An augmented value difference measure, Pattern Recognition Letters, 34(10), 1169–1174, (2013).

[7] Wilson, D. R. and Martinez, T. R. Improved heterogeneous distance functions, Journal of Artificial Intelligence Research, 6, 1–34, (1997).

[8] Li, C., Jiang, L., Li, H., Wu, J., and Zhang, P. Toward value difference metric with attribute weighting, Knowledge and Information Systems, 50, 795–825, (2017).

[9] Li, C., Jiang, L., and Li, H. Local value difference metric, Pattern Recognition Letters, 49, 62–68, (2014).

[10] Li, C., Jiang, L., Li, H., and Wang, S. Attribute weighted value difference metric, In: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, 575–580, (2013).

[11] Li, C. and Li, H. One dependence value difference metric, Knowledge-Based Systems, 24(5), 589–594, (2011).

[12] Li, Y., Fan, X., and Gaussier, E. Supervised categorical metric learning with schatten p-norms, IEEE Transactions on Cybernetics, 52(4), 2059–2069, (2020).

[13] Gu, D., Liang, C., and Zhao, H. A case-based reasoning system based on weighted heterogeneous value distance metric for breast cancer diagnosis, Artificial Intelligence in Medicine, 77, 31–47, (2017).

[14] Li, C. and Li, H. Selective value difference metric, Journal of Computers, 8(9), 2232–2238, (2013).

[15] Li, C., Jiang, L., and Li, H. Naive bayes for value difference metric, Frontiers of Computer Science, 8, 255–264, (2014).

[16] Skowron, A. and Wojna, A. K nearest neighbor classification with local induction of the simple value difference metric, In: Rough Sets and Current Trends in Computing: 4th International Conference, RSCTC 2004, Uppsala, Sweden, June 1-5, 2004, 229–234, (2004).

[17] Ortakaya, A. F. Independently weighted value difference metric, Pattern Recognition Letters, 97, 61–68, (2017).

[18] Liu, F., Vanschoenwinkel, B., Chen, Y., and Manderick, B. A modified value difference metric kernel for context-dependent classification tasks, In: 2006 International Conference on Machine Learning and Cybernetics, 3432–3437, (2006).

[19] Wilson, D. R. and Martinez, T. R. Value difference metrics for continuously valued attributes, In: Proceedings of the International Conference on Artificial Intelligence, Expert Systems and Neural Networks, 11–14, (1996).

[20] Rodriguez, Y., De Baets, B., Garcia, M. M., Morell, C., and Grau, R. A correlation-based distance function for nearest neighbor classification, In: Iberoamerican Congress on Pattern Recognition, 284–291, (2008).

[21] Kurian, M. J. and Gladston, R. S. An analysis on the performance of a classification based outlier detection system using feature selection, International Journal of Computer Applications, 132(8), 15–21, (2015).

[22] Townsend, J. T. Theoretical analysis of an alphabetic confusion matrix, Perception & Psychophysics, 9, 40–50, (1971).

[23] Cramer, Harald. Mathematical methods of statistics, Princeton University Press, 1999.

[24] Denis, Daniel J. Applied univariate, bivariate, and multivariate statistics: Understanding statistics for social and natural scientists, with applications in SPSS and R, John Wiley & Sons, 2021.

[25] Jiang, L. and Li, C. An augmented value difference measure, Pattern Recognition Letters, 34(10), 1169–1174, (2013).

[26] Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences, 55(1), 119–139, (1997).

[27] Demir, S. and Sahin, E. K. An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoost, Neural Computing and Applications, 35(4), 3173–3190, (2023).

[28] Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., and O'Sullivan, J. M. A review of feature selection methods for machine learning-based disease risk prediction, Frontiers in Bioinformatics, 2, 927312, (2022).