Current Science
CS 6 (1), 5864-5874 (2026)
Received 28 **December** 2025 | **Accepted** 12 **February** 2026 | **Online Available** 15 **February** 2026
https://doi.org/10.52845/CS/2026-6-1-17

Current Science
Current Science

OPEN ACCESS **ISSN(O) 2795-8639**

Original Article

Redundancy Reduction in Composite Indicator Systems for Causal Graph Construction

Han Lin[1], Ming Ji[1],*

[1]Academy of Military Science

***Corresponding Author: Ming Ji**

Abstract:
Indicator redundancy is common in capability assessments and, if uncontrolled, can bias results, reduce model performance, and distort causal graph learning. Using simulated datasets with known causal structures, this study examines how redundancy affects causal graph accuracy, complexity, stability, and model interpretability. We introduce MB-GAR, a Markov-blanket-guided genetic algorithm that removes redundant indicators while preserving information. Experiments show that MB-GAR markedly improves causal structure accuracy and predictive performance, outperforming correlation-based baselines by about 15% in redundancy-detection F1, demonstrating its effectiveness for building cleaner and more reliable evaluation indicator systems.
Keywords: comprehensive evaluation; causal graph; indicator redundancy; indicator system; Markov blanket

Introduction

In evaluating complex systems, composite indicator frameworks with multiple metrics are widely used to quantify performance. However, such systems often suffer from excessive redundancy, where many indicators are repetitive or highly correlated. Redundancy causes several issues: duplicate weighting that biases evaluation results [1]; reduced interpretability because correlated indicators obscure independent contributions [2]; higher data collection and processing costs without added information; and potential noise that lowers accuracy. In causal structure inference, redundancy may also create spurious links, producing inflated graphs and reducing interpretability.

Research highlights the need to control indicator redundancy. The OECD Handbook advises avoiding highly correlated indicators to prevent double-counting [3]. Large indicator systems often exhibit redundancy, reducing comparability and scientific validity. Thus, designers commonly apply correlation or factor analysis to remove redundant indicators. Recent work introduces advanced methods, such as MIC for detecting correlated clusters [4] and causal-based optimization approaches [5].

From a data-driven perspective, feature selection research emphasizes removing redundant features to improve performance. mRMR selects features highly relevant yet minimally redundant [6]. Surveys show redundancy harms generalization and should be addressed with filter or wrapper methods [7]. However, in capability evaluation, indicators carry structural or semantic roles, so correlation-based deletion may discard key information [8]. Thus, redundancy effects must be quantified and controlled to ensure accuracy and robustness in practical system assessments contexts broadly.

Machine learning models are widely used in evaluations. Ensemble methods such as Random Forests [9] and XGBoost [10] perform well and include mechanisms that partly mitigate redundancy—RF via random feature splits, XGBoost via regularization. Yet redundant features still impair models; studies show RF accuracy declines when redundant variables are present [11, 12]. Because ensemble base learners must be accurate and independent, excessive correlation weakens performance. Thus, comparing model behavior under indicator redundancy remains essential.

This study analyzes how indicator redundancy affects capability evaluation accuracy and stability. We simulate data with controllable redundancy, train Random Forest and XGBoost models, and assess performance changes. We also apply PC [13], GES [14], and MMHC [15] to examine redundancy's impact on causal graph accuracy, complexity, and stability. We focus on several aspects:

1. The impact of redundant indicators on model prediction error and result stability;
2. The manifestations of redundancy in the indicator network structure, such as changes in the indicator correlation network and causal graph structure;
3. The effect of redundancy on model feature importance and feature selection results.
4. The influence of redundancy on model generalization ability, including the model's tendency to overfit.

We offer quantitative guidance for designing indicator systems that control redundancy while preserving comprehensiveness. We also introduce MB-GAR, a redundancy-filtering method used before causal graph construction, and validate its effectiveness in identifying redundant indicators and balancing information retention.

2. Methods

2.1 Data Simulation and Indicator Redundancy Modeling

To analyze indicator redundancy under controlled conditions, we adopt a simulation framework. We first generate base indicators representing independent key metrics. Let the number of base indicators be d, denoted $X_1, X_2, ..., X_d$. Each X_i is independently sampled from a standard normal distribution ($X_i \sim N(0,1)$), ensuring independence. We then introduce redundant indicators to expand the system. For each base indicator X_i, a redundant indicator is constructed by adding small random noise: $X_{i'} = X_i + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$ with small variance. By tuning σ, we control the correlation between $X_{i'}$ and X_i: smaller σ makes this correlation approach 1, representing higher redundancy. In this study, we choose σ so that the correlation is approximately 0.95, mimicking a high-redundancy setting. Repeating this process yields multiple redundant indicators per base variable. Let the total number be r. Redundancy level is reflected by r or by the proportion of redundant indicators. When $r = 0$, base indicators exist; $r > 0$ introduces redundancy. We examine scenarios $r = 0, 5, 10, 15, 20$. As illustrated in Figure 1, higher redundancy produces clusters of highly correlated indicators linked by thick edges, with weak links between clusters, making the network denser and less independent.

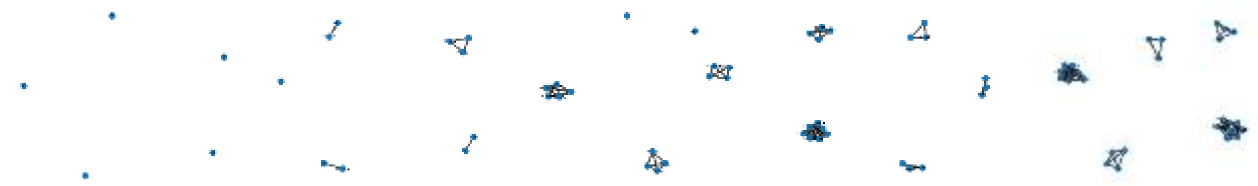


Figure 1. causal graph structure variations under different numbers of redundant indicators r: (a) r=0; (b) r=5; (c) r=10; (d) r=15; (e) r=20;

Next, we simulate the composite capability score Y as the prediction target. We assume capability depends mainly on base indicators, i.e., $Y = \sum_{j=1}^{d} w_j X_j + \xi$, where w_j denotes each indicator's contribution and ξ is independent noise. The weight vector $\mathbf{w}=(w_1,\ldots,w_d)$ is randomly generated to avoid symmetry. The noise $\xi \sim N(0,\sigma^2)$ is set so that 80–90% of Y's variance comes from the signal, yielding a high signal-to-noise ratio while preserving realistic uncertainty.

These steps yield a dataset with known redundancy: each of the n samples contains $(d+r)$ indicators and a composite score Y. The true causal graph is also known—each X_j affects Y and is the sole parent of its redundant $X_{j'}$, with no cross-cluster causal links.

2.2. Model Selection and Evaluation Metrics

We use Random Forest (RF) and XGBoost as evaluation models because they perform well in prediction tasks [10] and differ in how they handle correlated or redundant features. RF builds many trees using bootstrap samples and random feature subsets, which helps reduce multicollinearity. However, when redundancy is high, trees may split arbitrarily on equivalent features, generating uninformative splits and diluting feature importance. Gregorutti et al. showed that highly correlated features in RF weaken each other's importance, obscuring truly influential variables and complicating feature selection [12]. Thus, RF remains sensitive to extensive redundancy.

XGBoost, a gradient-boosted decision tree method, trains trees sequentially to fit residuals and improve accuracy. Its strong performance comes from second-order optimization, incremental training, and regularization such as L_1/L_2 penalties and subsampling. Although it does not discard features, regularization helps limit overfitting to redundant variables. Yet highly correlated features may still be repeatedly reused: a first tree may split on X_1, and later trees may switch to redundant $X_{1'}$ for similar gains, accumulating duplicate information. This raises model complexity, slightly harms accuracy by fitting noise, and increases training time and memory use.

We evaluate model accuracy using R^2 and RMSE. R^2 measures goodness of fit, with values near 1 indicating strong variance explanation. RMSE quantifies prediction error magnitude, where smaller values denote higher precision.

2.3. Causal Structure Learning and Evaluation

To learn causal relationships among indicators, we employ three representative structure learning algorithms: the constraint-based PC algorithm, the score-based Greedy Equivalence Search (GES), and the hybrid MMHC. We evaluate the learned graphs against the ground-truth structure using four metrics.

1. Structural accuracy: precision, recall, and F1 score of correctly identified edges. Precision measures the proportion of learned edges that are true; recall measures the proportion of true edges recovered; F1 is their harmonic mean.
2. Structural Hamming distance (SHD): the number of edge additions, deletions, or reversals required to convert the learned graph into the true one; lower SHD indicates higher fidelity.
3. Structural complexity: quantified by total edges and average node degree. Higher values imply denser, more complex graphs. We also track redundant paths, such as when a base indicator and its redundant copies all point to the same target Y, inflating its parent set.
4. Structural stability: assessed by repeatedly learning the structure under identical conditions and computing Jaccard similarity between edge sets. High variance in Jaccard values indicates instability, while low variance reflects consistent structures.

2.4. Markov Blanket-Guided Genetic Algorithm (MB-GAR) for Redundancy Elimination

To automatically identify and eliminate redundant indicators, we propose MB-GAR, which integrates Markov blanket theory with a genetic algorithm. Its goal is to reduce redundancy while preserving maximal information. The method has two stages: first, identify each indicator's Markov blanket to determine conditional dependencies; second, construct an optimization objective and use a genetic algorithm to search for an indicator subset that minimizes redundancy.

1. Markov Blanket Theoretical Basis: The Markov blanket (MB), introduced by Pearl [16], is the set of variables that renders a node

statistically independent of all others when conditioned on it. For any indicator in a Bayesian network, its MB includes its parents, its children, and the other parents of its children ("spouses"). This boundary represents the minimal sufficient information set: once an indicator's MB is known, the indicator contributes no additional information beyond what the MB already provides. Thus, if an indicator's information is fully captured by its MB, it is redundant. In feature selection, MBs are considered "local causal" structures because identifying them does not require reconstructing the full causal network; only direct conditional dependencies must be found. Lamsaf et al. noted that correlation-based feature selection often misses causal relationships, whereas incorporating causal reasoning improves interpretability and robustness [17]. When domain knowledge exists, it can directly guide MB identification, reducing uncertainty.

2. Optimization Objective for Redundancy Identification: We define a "redundant indicator" as one that, given other indicators, contributes no new independent information or whose information can be inferred from others. This follows Pan et al.'s objective of selecting subsets with maximum relevance and minimum redundancy [18]. Accordingly, we quantify redundancy and information retention to construct an optimization objective.

- Redundancy: Indicators with highly similar or overlapping Markov blankets influence the evaluation through similar pathways, making their information duplicative. Redundancy for a subset is thus measured via pairwise MB overlaps. High redundancy indicates extensive overlap.

- Information Retention: A good subset should preserve as much information from the full set as possible. For each excluded indicator, we identify the selected indicator whose MB has the highest overlap with it and use this maximum value. Averaging these across all excluded indicators yields the subset's information coverage, reflecting how well the retained indicators "explain" the discarded ones.

Based on these two criteria, we construct an objective that maximizes information retention while minimizing redundancy. A tuning parameter λ balances the trade-off by penalizing redundancy.

3. Genetic Algorithm for Optimal Subset Search: Selecting the optimal subset from N indicators is a combinatorial optimization problem with a search space of size 2^N, making exhaustive search infeasible for large N. We therefore employ a Genetic Algorithm (GA), which simulates natural selection to evolve candidate subsets. GA's population-based search provides strong global exploration and avoids local optima. Subsets are encoded as binary vectors, enabling efficient crossover and mutation. The GA iteratively evaluates fitness, selects superior individuals, performs crossover and mutation, and converges toward an optimal low-redundancy, high-information subset.

- Encoding: Each candidate subset is encoded as a binary chromosome of length N, where the j-th bit is 1 if the j-th indicator is selected and 0 otherwise. For example, "11001000" (for $N=8$) selects indicators 1, 2, and 5.

- Initial Population: A diverse set of randomly generated chromosomes forms the first generation, ensuring broad search coverage.

- Fitness Evaluation: Each chromosome's fitness is computed using the objective function, where higher fitness reflects a better balance between information retention and reduced redundancy.

- Selection, Crossover, Mutation: High-fitness individuals are chosen as parents. Crossover exchanges chromosome segments to create new subsets, while mutation flips bits with a small probability to maintain diversity and avoid premature convergence.

- Iteration and Termination: The evaluate–select–crossover–mutate cycle repeats until reaching a generation limit or convergence. The chromosome with the highest fitness is selected as the optimal reduced indicator set; unselected indicators are treated as redundant.

This GA-based search yields an indicator set that balances redundancy and information retention without requiring subjective thresholds [18]. By modeling multivariate redundancy globally, it outperforms simple correlation filtering. We then apply MB-GAR to simulated data to evaluate its effectiveness.

3. Experimental Design and Procedure

Based on the methodology above, we designed an experimental procedure to systematically assess the impact of indicator redundancy and evaluate redundancy-elimination strategies.

1. Redundancy scenario setup: We generate simulated datasets with predefined numbers of redundant indicators: 0, 5, 10, 15, and 20. The number of base indicators is fixed at 5 (representing performance, efficiency, quality, safety, and cost), giving total indicator counts of 5, 10, 15, 20, and 25. Each scenario includes sufficient samples for reliable model training and evaluation.
2. Dataset splitting: Each dataset is split into 70% training and 30% test data. All models and causal-structure learning algorithms are trained on the training set. For every redundancy level, we repeat data splitting and training 10 times to assess stability.
3. Model training: In each repeat, we train Random Forest and XGBoost models to predict the composite score. Standard hyperparameters highlight redundancy effects rather than tuning: RF uses 100 trees; XGBoost uses 100 boosting rounds, learning rate 0.1, and depth 6. Both achieve near-optimal fit under no redundancy.
4. Causal graph structure learning: For each scenario, PC, GES, and MMHC are applied to learn causal structures, with multiple runs to evaluate consistency.
5. Model performance evaluation: For each repeat, we record R^2 and RMSE, then compute mean and standard deviation across runs.
6. Result analysis: To understand redundancy effects more deeply, we conducted several analyses.

- Indicator network structure: For each scenario, we computed the indicator correlation matrix and built a correlation network. Changes in average degree, clustering coefficient, and other topological properties illustrate how redundancy increases network density and reduces indicator independence.
- Causal graph structural accuracy: Using the known ground truth, we compared learned graphs with the true structure, evaluating precision, recall, F1, and Structural Hamming Distance (SHD). We also recorded graph complexity (average degree, redundant paths) and stability (variance of Jaccard similarity across runs) to assess how redundancy distorts causal structures.
- Feature importance: We extracted importance measures from Random Forest and XGBoost. Without redundancy, base indicators rank clearly; under high redundancy, multiple redundant features split the original importance, obscuring truly critical indicators.
- Error distribution: We plotted prediction error distributions to observe changes in shape, variance, and tail behavior, quantifying how redundancy affects reliability.
- Model comparison: We compared the sensitivity of RF and XGBoost to redundancy. Larger accuracy drops or stability changes indicate greater vulnerability, highlighting differences in their handling mechanisms.
- Redundancy elimination test: Under the highest redundancy (20 redundant indicators), we applied MB-GAR to identify and remove redundant indicators. After learning Markov blankets and optimizing via GA, the reduced subset was used to retrain the models and reassess causal structures. We also applied a correlation-threshold baseline (correlation > 0.9) for comparison. Contrasting results demonstrate MB-GAR's superiority in eliminating redundancy while preserving information.

All experiments were repeated independently to ensure statistical robustness.

4. Results and Analysis

4.1. Relationship between Causal Graph Structural Accuracy and Redundancy Level

Table 1 summarizes how redundancy affects causal graph accuracy and complexity. With no redundant indicators, all algorithms accurately recover the true structure: edges closely match the real relations, F1 scores approach 1.0, SHD is near zero, and node degree averages about 1.6. As redundancy increases, accuracy declines sharply. With 10 redundant indicators, F1 scores fall to around 0.6; with 20, they drop to roughly 0.4. SHD rises steeply, indicating many incorrect or missing edges.

Under high redundancy, each base indicator and its redundant copies often connect to the same target, producing multiple redundant paths and substantially increasing node degrees. Some algorithms incorrectly infer direct causal links among redundant indicators, mistaking shared causes for causal interactions. Overall graph complexity rises: average degree increases from 1–2 (no redundancy) to 5–6 (high redundancy), yielding much denser networks.

Redundancy also reduces structural stability. Across repeated runs, high-redundancy graphs vary widely: the average Jaccard similarity between edge sets drops from nearly 1.0 (no redundancy) to about 0.5. Thus, redundancy makes learned causal structures inaccurate, overly complex, and unstable, causing them to diverge significantly from the true network.

Table 1. Causal graph structure learning performance under different levels of redundant indicators.

Number of Redundant Indicators	Method	F1	SHD	Avg. degree	Avg. Jaccard
0	GES	0.989	0.100	1.660	0.960
	MMHC	0.944	0.500	1.633	0.809
	PC	1.000	0.000	1.667	1.000
5	GES	0.644	9.400	2.982	0.535
	MMHC	0.631	9.300	2.805	0.466
	PC	0.644	9.400	2.982	0.535
10	GES	0.485	25.000	4.200	0.518
	MMHC	0.488	24.500	4.112	0.502
	PC	0.485	25.000	4.200	0.518
15	GES	0.431	42.900	5.267	0.486
	MMHC	0.430	42.100	5.133	0.461
	PC	0.431	42.900	5.267	0.486
20	GES	0.402	63.800	6.262	0.550
	MMHC	0.405	63.000	6.185	0.533
	PC	0.402	64.400	6.338	0.567

4.2. Relationship between Model Performance and Redundancy Level

Table 2 shows that with no redundancy, Random Forest and XGBoost achieve high accuracy ($R^2 \approx 0.863$ and 0.883). As redundant indicators increase, accuracy declines sharply, reaching about 0.829 (RF) and 0.855 (XGB) at 10 redundants. At higher redundancy levels (15–20), accuracy partially rebounds but remains below baseline; with 20 redundants, RF reaches 0.843 and XGB 0.865. This pattern reflects a clear drop followed by slight recovery as redundancy intensifies.

Table 2. Performance metrics of Random Forest and XGBoost under different numbers of redundant indicators (mean ± standard deviation).

Number of Redundant Indicators	RF R^2	XGB R^2	RF RMSE	XGB RMSE
0	0.863 ± 0.017	0.883 ± 0.016	0.891 ± 0.073	0.821 ± 0.054

5	0.856 ± 0.021	0.879 ± 0.020	0.891 ± 0.085	0.816 ± 0.078
10	0.829 ± 0.014	0.855 ± 0.024	0.949 ± 0.076	0.871 ± 0.064
15	0.831 ± 0.021	0.856 ± 0.029	0.976 ± 0.130	0.892 ± 0.097
20	0.843 ± 0.017	0.865 ± 0.020	0.984 ± 0.116	0.911 ± 0.105

[1] Each scenario's results are averaged over 10 independent runs; higher R^2 is better, lower RMSE is better.

Figure 2 shows that model R^2 declines as redundancy increases, forming a U-shaped pattern with an initial drop and slight rebound. Although redundant indicators add no new information, finite data and model complexity cause accuracy loss. Importantly, redundancy does not affect output stability: Table 2 shows nearly unchanged standard deviations across scenarios—RF's R^2 std remains ~0.017 and XGBoost's rises only from ~0.016 to ~0.020 at 20 redundants. The shaded ±1-std bands in Figure 2 confirm that variability is nearly constant. Thus, redundancy mainly reduces mean accuracy while leaving prediction stability largely unaffected.

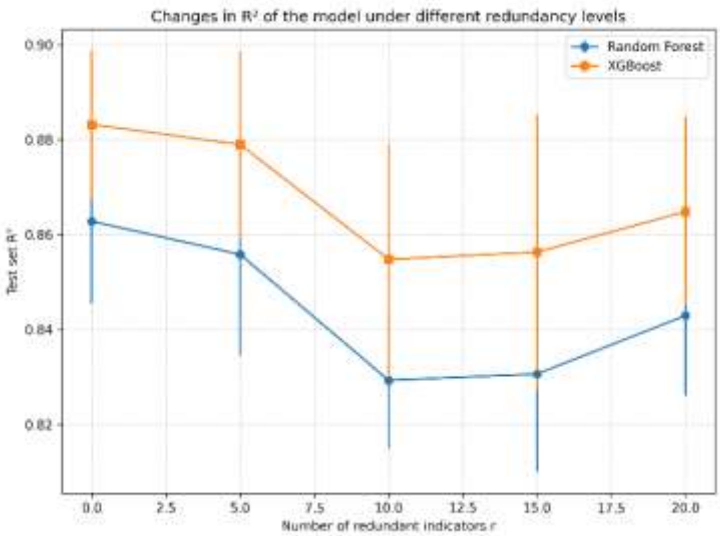


Figure 2. R^2 **of Random Forest and XGBoost models under different numbers of redundant indicators (shaded area denotes ±1 std).**

Random Forest and XGBoost show similar trends under redundancy: performance drops most at moderate redundancy and rebounds slightly at very high levels. Random Forest is more affected—for 10 redundant indicators, its R^2 decreases by 0.034, compared with XGBoost's 0.028, and its RMSE rises more sharply. This reflects RF's random feature selection, which repeatedly draws features from the same redundant cluster, amplifying noise. XGBoost's sequential residual fitting and strong regularization help maintain focus on informative features, making it slightly more robust to redundancy.

4.3. Impact of Redundancy on Feature Importance

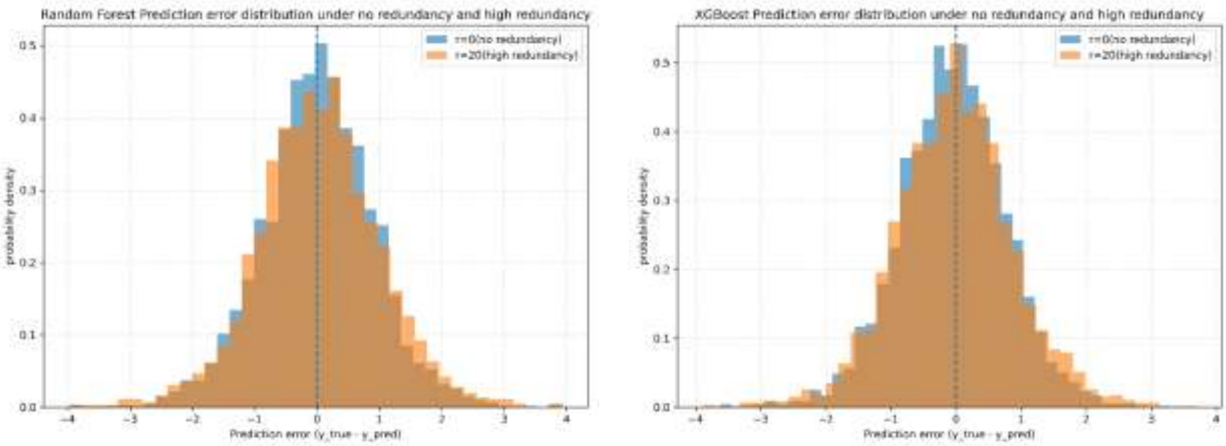
As redundancy increases, the learned causal graph changes markedly (Figure 1). High-redundancy scenarios produce clusters of highly correlated indicators that jointly influence the same target, sharply increasing average node degree and clustering coefficients. These dense substructures reduce indicator independence and create groups of "equivalent information." Such complexity complicates visualization and interpretation because many correlated indicators point to the same target, generating redundant causal paths. This suggests the need to remove or merge redundant nodes before graph construction.

Redundancy similarly affects downstream predictive models. In Random Forests, with no redundancy, the five base indicators have clearly distinguishable importance. Under high redundancy, however, each base indicator's importance is diluted by its redundant copies: a feature contributing ~20% in the baseline may drop below 10% when redundant variants are alternately selected, obscuring its true role. XGBoost shows the same dilution pattern, repeatedly selecting equivalent features across many trees.

This dilution weakens the ability of importance-based feature selection to identify key indicators, causing mis-ranking of truly influential variables. Consequently, redundancy must be controlled systematically—e.g., using MB-GAR—before constructing causal graphs to prevent distorted pathways and unreliable feature-importance signals.

4.4. Impact of Redundancy on Error Distribution and Generalization

To visualize redundancy effects on prediction errors, we compared error distributions for no-redundancy and high-redundancy scenarios (Figure 3). For Random Forest, the no-redundancy distribution is bell-shaped around 0 with most errors within ±3. Under high redundancy, the peak and range remain similar, but the spread widens slightly, indicating increased variability without more extreme errors. XGBoost shows the same pattern: modestly larger dispersion but unchanged shape and tail behavior. Thus, redundancy mainly lowers average accuracy rather than increasing instability or extreme mistakes. Practically, this means redundancy weakens model fit but does not significantly raise worst-case risk; removing redundancy chiefly improves mean predictive accuracy.



(a) (b)

Figure 3. Comparison of model prediction error distributions under no-redundancy vs. high-redundancy conditions: (a) random forest prediction error distributions under no-redundancy vs. high-redundancy conditions; (b) XGBoost prediction error distributions under no-redundancy vs. high-redundancy conditions

We examined whether redundancy increases overfitting, which occurs when a model fits training data well but performs poorly on test data. With moderate model complexity and noise in our simulations, redundant features alone did not greatly worsen overfitting. However, when complexity constraints were removed—e.g., deeper XGBoost trees or unrestricted Random Forests—high redundancy caused models to learn spurious noise patterns. With 20 redundant indicators, XGBoost achieved near-perfect training R^2 but much lower test R^2, showing clear overfitting. Redundancy thus raises overfitting risk by adding useless dimensions unless sufficient regularization or data is provided.

4.5. Effects of Redundancy Elimination: MB-GAR Method Evaluation

In the 20-redundancy scenario, MB-GAR selected 7 indicators from 25, effectively retaining one representative per redundant cluster. All 5 base indicators were preserved, and only 2 redundant ones remained due to complementary information, showing that most redundant features were correctly identified and removed.

Table 3. Comparison of model performance and causal structure learning results before redundancy removal vs. after applying MB-GAR (with no-redundancy baseline for reference).

Category	Metric	Before Removal	After MB-GAR	No-Redundancy Baseline
Model Performance	Random Forest R^2	0.843	0.860	0.863
	XGBoost R^2	0.865	0.880	0.883
	Random Forest RMSE	0.984	0.891	0.891
	XGBoost RMSE	0.911	0.821	0.821
	XGBoost R^2 std	0.020	0.017	0.016
Causal Structure	Structural F1	0.40	0.95	≈1.0
	SHD	>60	0–1	0
	Jaccard variance	High	Very low	Very low
Redundancy Identification	Precision	–	100%	–
	Recall	–	90%	–
	F1 Score	–	0.947	–

Table 3 shows that after applying MB-GAR, Random Forest's R^2 improves from 0.843 to 0.860 and XGBoost's from 0.865 to 0.880, nearly restoring no-redundancy performance. RMSE likewise decreases for both models. Stability also improves: in the 20-redundancy case, XGBoost's R^2 standard deviation drops from 0.020 to ~0.017, indicating reduced sensitivity to data perturbations and more robust predictions.

Beyond improving prediction, MB-GAR greatly enhanced causal structure learning. In high-redundancy data, learned graphs contained many spurious edges, but retaining only one representative per cluster produced far simpler, cleaner structures. With PC, a node's ~10 parents dropped to about 5, and erroneous edges nearly vanished. F1 accuracy rose from ~0.40 to >0.95, and SHD fell from over 60 to 0–1. Stability also improved markedly, with Jaccard variance decreasing by an order of magnitude.

To assess MB-GAR's redundancy identification ability, we compared its outputs with the simulation ground truth. Among 20 truly redundant indicators, MB-GAR correctly identified 18, retaining 2 because their partially distinct Markov blankets provided complementary information. This yields 100% precision, 90% recall, and an F1 of ~0.947. A correlation-threshold baseline (0.9) achieved only ~0.82 F1, mistakenly removing indicators with moderately high but informative correlations and thus losing information. MB-GAR therefore identifies redundancy more accurately while preserving coverage. Similar advantages were reported by Cheng et al., where a Markov blanket–based GA improved redundancy-reduction F1 by about 13% over MIC-MAC [19].

In summary, MB-GAR effectively reduces redundancy, improves model accuracy and stability, and greatly enhances causal structure learning. Unlike simple correlation filtering, it identifies conditional redundancies while preserving essential information, simplifying the indicator system without sacrificing reliability.

5. Discussion

This study examined how indicator redundancy affects system capability evaluation and how redundant indicators can be effectively reduced. Using simulated data with controllable redundancy and known causal structures, we quantified redundancy's impact on model accuracy, stability, and causal graph integrity. Results show that uncontrolled redundancy produces spurious causal edges, inflates graph complexity, and reduces interpretability. It also lowers average predictive accuracy, though it has limited influence on variance or extreme errors. To address these issues, we proposed MB-GAR, which integrates Markov blanket theory with a genetic algorithm to automatically identify and eliminate redundant indicators. Experiments demonstrate that MB-GAR sharply reduces indicator count with minimal information loss, improves predictive performance, and greatly enhances causal graph accuracy and stability. Compared with correlation-threshold filtering, MB-GAR provides more comprehensive and accurate redundancy detection.

These findings offer practical guidance for designing evaluation indicator systems: redundancy must be controlled to prevent bias and interpretation difficulties. Redundancy reduction is especially important before causal graph construction to avoid spurious relationships. MB-GAR shows that combining causal inference with intelligent optimization achieves a balance between information completeness and parsimony. Future work will validate these results on real data, explore other redundancy forms, and integrate MB-GAR with advanced learning techniques.

References

1. Wehbe, C. and Baroud, H., Limitations and considerations of using composite indicators to measure vulnerability to natural hazards. Sci. Rep., 2024, 14(1), 19333.
2. Brint, A., Genovese, A., Piccolo, C. and Settanni, E., Reducing data requirements when selecting key performance indicators for supply chain management: the case of a multinational automotive component manufacturer. Int. J. Prod. Econ., 2021, 233, 107967.
3. European Union and Joint Research Centre, Handbook on Constructing Composite Indicators: Methodology and User Guide, OECD Publishing, Paris, France, 2008, pp. 1–160.
4. Wu, L., Pan, Y., Chen, K. et al., Analysis of urban water-saving driving factors based on maximum information coefficient. People's Yellow River, 2023, 1, 87–92.
5. Fan, B., Zhong, J., He, Y. et al., A causality-based evaluation index system optimization method and system. China Patent CN202311095124.8, 2023.
6. Peng, H., Long, F. and Ding, C., Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell., 2005, 27(8), 1226–1238.
7. Chandrashekar, G. and Sahin, F., A survey on feature selection methods. Comput. Electr. Eng., 2014, 40(1), 16–28.
8. Zhang, J., Li, S., Yang, H. et al., Efficient and intelligent feature selection via maximum conditional mutual information for microarray data. Appl. Sci., 2024, 14(13), 1–15.
9. Cutler, A., Cutler, D. R. and Stevens, J. R., Random forests. In Ensemble Machine Learning (eds. Zhang, C. and Ma, Y.), Springer, New York, USA, 2012, pp. 157–175.
10. Chen, T. and Guestrin, C., XGBoost: a scalable tree boosting system. In Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, San Francisco, USA, 13–17 August 2016.
11. Kubus, M., The problem of redundant variables in random forests. Acta Univ. Lodz. Folia Oecon., 2018, 6(339), 7–16.
12. Gregorutti, B., Michel, B. and Saint-Pierre, P., Correlation and variable importance in random forests. Stat. Comput., 2017, 27(3), 659–678.
13. Spirtes, P. and Glymour, C., An algorithm for fast recovery of sparse causal graphs. Soc. Sci. Comput. Rev., 1991, 9(1), 62–72.
14. Chickering, D. M., Optimal structure identification with greedy search. J. Mach. Learn. Res., 2002, 3, 507–554.
15. Tsamardinos, I., Brown, L. E. and Aliferis, C. F., The max–min hill-climbing Bayesian network structure learning algorithm. Mach. Learn., 2006, 65(1), 31–78.
16. Pearl, J., Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, USA, 1988, pp. 1–500.
17. Lamsaf, A., Carrilho, R., Neves, J. C. et al., Causality, machine learning, and feature selection: a survey. Sensors, 2025, 25(8), 2373.
18. Pan, X., Sun, J., Yu, H. et al., Feature selection using non-dominant features-guided search for gene expression profile data. Complex Intell. Syst., 2023, 9(6), 6139–6153.
19. Cheng, X., Shi, H. and Dong, Y., Optimization

of farmers' credit evaluation index system based on MIC-MAC. J. Shanxi Univ. Finance Econ., 2018, 48(8), 17–25.