ISSN 2723-6471

An Effective Hybrid Approach for Predicting and Optimizing Business Complexity Metrics and Data Insights

Rahmad B.Y Syah[1,*], Marischa Elveny[2], Rana Fathinah Ananda[3], Mahyuddin K.M Nasution[4], Hartono[5]

[1,5]*Excellent Centre of Innovations and New Science-PUIN, Faculty of Engineering, Universitas Medan Area, Medan, 20223. Indonesia*

[2,4]*Fasilkom-TI, Universitas Sumatera Utara, Medan. 20115. Indonesia*

[3]*Faculty of Economics and Business, Universitas Medan Area. Medan. Indonesia*

(Received: December 22, 2024; Revised: January 30, 2025; Accepted: April 30, 2025; Available online: July 19, 2025)

Abstract

This study proposes a hybrid approach for optimizing complexity prediction in the domain of business intelligence by integrating three powerful techniques: the Multi-Objective Complexity Prediction Model (MPK), Principal Component Analysis (PCA), and the XGBoost regression algorithm. The MPK model serves as a state-based simulator to capture system complexity dynamics, while PCA is employed to reduce data dimensionality and eliminate redundancy among features. Subsequently, XGBoost is used as a non-linear predictive model to estimate complexity values based on the refined input features. The results show that this hybrid approach significantly improves prediction accuracy, reduces data noise, and streamlines the modelling process. Quantitative evaluation using Mean Squared Error (MSE), Mean Absolute Error (MAE), and the R-squared (R^2) metric demonstrates exceptional performance, with an MAE of 0.000035, an MSE of 6.7×10^{-9}, and an R^2 of 0.9999999. These results confirm that the integration of MPK, PCA, and XGBoost is highly effective for complexity prediction tasks and can provide accurate and insightful outcomes in business intelligence analytics.

Keywords: Hybrid Models, MPK, PCA, Xgboost, Multi-Objective Optimization

1. Introduction

This In a digital era characterized by exponential data growth, the ability to analyze and extract insights from business data (business intelligence/BI) becomes a key factor in strategic decision-making [1], [2]. Business data is not only complex and dynamic but also contains high uncertainty that is difficult to accommodate with conventional prediction models. As the volume, variety, and velocity of data increase, organizations need a more adaptive and high-precision approach to handling information to support business resilience and growth [3].

Various machine learning approaches, such as regression, SVM, and neural networks, have been used for business prediction, but they still have limitations in handling high-dimensional data and non-linear relationships between variables [4], [5]. Previous studies by [6], [7] emphasized the importance of integrating big data analytics and machine learning to strengthen prediction strategies, while [8] demonstrated the effectiveness of a hybrid approach in multi-input systems.

However, there are still few studies that simultaneously combine state-space-based dynamic simulation models such as the Multi-Objective Complexity Prediction Model, dimensionality reduction techniques such as Principal Component Analysis, and boosting algorithms such as Extreme Gradient Boosting [9], [10], [11]. In this approach, MPK is used to simulate the dynamics of system complexity through a state-space representation that combines input, uncertainty, and output factors. PCA plays a role in reducing the dimensionality of raw data so that the main variance is maintained and noise can be minimized; while XGBoost builds a robust non-linear prediction model through a decision tree-based boosting technique [12], [13]. Therefore, this study proposes a hybrid MPK–PCA–XGBoost

***Corresponding author: Rahmad B.Y Syah (rahmadsyah@uma.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i3.830

approach that aims to simplify data structures, strengthen prediction processes, and generate strategic insights into business data to support more accurate and adaptive data-driven decision making.

2. Literature Review

The rapid expansion of business data, coupled with the rising demand for intelligent decision-making, has led to a surge in research focused on hybrid models for prediction and optimization. These hybrid approaches aim to address the growing complexity of business metrics by combining multiple analytical paradigms such as machine learning, optimization algorithms, and interpretability frameworks into a unified predictive structure.

Conducted a comprehensive systematic literature review covering hybrid methods that integrate machine learning with optimization techniques. Their work highlights the growing adoption of such methods in domains requiring high accuracy and adaptability, emphasizing that combining metaheuristics (e.g., genetic algorithms, PSO, simulated annealing) with ML can significantly improve prediction performance, especially under complex business scenarios [14].

Similarly, [15] explored the implementation of ensemble deep learning models integrated with genetic optimization techniques in financial forecasting. Their findings confirm that hybrid architectures improve predictive robustness and convergence while maintaining a manageable level of complexity a crucial aspect for business contexts where reliability and interpretability matter.

In real-world applications, complexity in business prediction often arises from dynamic environments, multidimensional metrics (e.g., time, cost, quality), and temporal patterns. A recent study combining XGBoost with Simulated Annealing demonstrated how optimization-enhanced models can outperform standard ML approaches in accurately forecasting project timelines and costs, proving that hybrid models can handle prediction complexity in uncertain operational settings [16].

Moreover, process mining has emerged as a valuable complement to ML in hybrid systems. By combining event log analysis with predictive models, organizations can identify bottlenecks and predict behavioural outcomes in operational workflows. This fusion, as proposed in hybrid process mining research, opens new paths to understanding consumer behaviour, operational inefficiencies, and business process optimization [17]. Research Gaps and Contributions to Hybrid Methods can be seen in table 1.

Table 1. Research Gap and Contribution to Hybrid Method

Research Gap	Supporting Reference	Contribution of This Study
Most hybrid models emphasize accuracy but overlook prediction complexity and its impact on decision-making.	B. F. Azevedo, A. M. A. C. Rocha, and A. I. Pereira [14]	Introduces a hybrid framework that explicitly targets prediction complexity in addition to improving accuracy.
The hybrid approach emphasizes financial forecasting, but indicators of business complexity and data insights remain unexamined.	X. Zhu [15]	Creating an innovative hybrid framework: delivering meaningful data insights.
Predictive Analytics and Machine Learning for Real-Time Supply Chain Risk Mitigation and Agility.	A. Aljohani [7]	Applies hybrid ML–SA models to predict complex operational KPIs with improved efficiency.
The financial models predominantly focus on single-criteria optimization, neglecting multi-objective optimization and dimensionality reduction for Business Intelligence (BI).	A. Jha, S. Maheshwari, P. Dutta, and U. Dubey [17]	Developing optimization techniques for enhancing predictive accuracy and providing insightful advice.

3. Methodology

This study uses a hybrid approach consisting of three main stages, starting with complexity simulation using the MPK to generate a system complexity score through a state-space representation that combines input, disturbance, and output [18], [19]. Furthermore, dimensionality reduction is carried out using PCA to filter out key features that have high

 ISSN 2723-6471

variance and reduce data noise [20]. Finally, the combination of MPK complexity scores and PCA result features is used as input to the XGBoost algorithm, which builds a non-linear prediction model based on boosting techniques [21]. Evaluation is carried out using MSE, MAE, and R^2 metrics to assess the accuracy and effectiveness of the model in the context of business intelligence data [22], [23], [24]. as shown in figure 1.

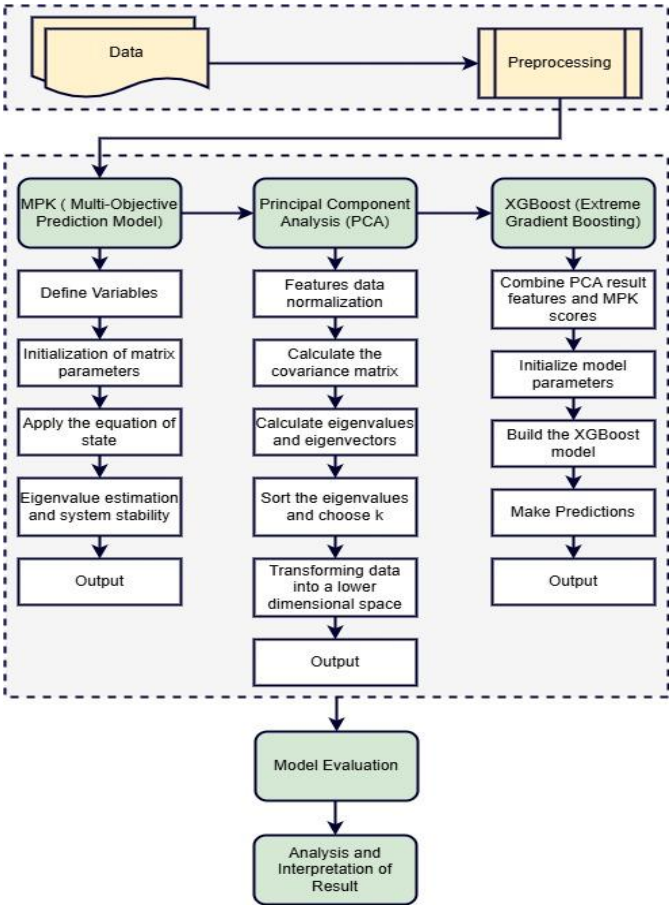

Figure 1. Research Methodology

3.1. Basic equation of MPK state

In this study, MPK uses a state-space approach as a mathematical basis for representing the dynamics of complex systems involving many input variables and disturbances [25]. This model assumes that the complexity of the system is not only influenced by the main input variables x, but also by disturbances or external variables u, and the relationship between these variables occurs dynamically over time [26].

$$x_{k+1} = A\,x_k + B\,u_k + E\,d_k \tag{1}$$

x_k : State Vector at Time k; u_k : Control Vector at Time k; d_k : Disturbance Vector at Time k; A_k, B_k, E : Parameter Matrix for State Models; C, D : Parameter Matrix for the Output Model

$A\,x_k$ It is the contribution of the previous state x_k which is "passed on" to x_{k+1} through the matrix A. If A is the identity matrix (with slight modifications), then most of the old state is directly carried over to the new state. $B\,u_k$ Captures the direct effect of the control/input variable u_k on changes in state. Equality:

$$MPK = \max\left(\sum_{i=1}^{n} W_i \times IOF_i(x_{k+1}) + \sum_{j=1}^{m} V_j \times UOF_j(x_{k+1}, A_k x_k) + \sum_{k=1}^{p} X_k \times OOF_k(y_k, Cx_k)\right) \tag{2}$$

 ISSN 2723-6471

x_{k+1} : State at the Next Time; $A_k x_k$: is the Previous State that ss Projected; y_k : State at the Next Time; $IOF_i(.)$: Input Optimization Factor; $UOF_j(.)$: Uncertainty Optimization Factor; OOF_k (.): Output Optimization Factor; W_i, V_i, X_k : Component Weights.

Cx_k Captures the contribution of the "state" x_k to the complexity score y_k For example, if x_k represents [system response, process load, resource efficiency, ...], then C determines how much each component adds up to the complexity score. After the MPK process (Equations (1)(2)) produces an output $y_k \epsilon \mathbb{R}$ (the complexity score at time k), and the PCA process (Equation (3) through selecting k components) yields a feature vector.

$$z_k = \left[PC_{t,1}, PC_{t,2}, \ , PC_{t,k}\right] \epsilon \mathbb{R}^k \tag{3}$$

The next step is to combine these two outputs into a single, unified feature vector $x_t^{combined}$.

$$x_t^{combined} = \begin{bmatrix} y_t \\ PC_{t,1} \\ PC_{t,2} \\ . \\ . \\ . \\ PC_{t,k} \end{bmatrix} \epsilon \mathbb{R}^{k+1} \tag{4}$$

3.2.Dimensionality Reduction with PCA

At this stage, PCA is used as a dimension reduction technique to simplify the data structure of the complex system generated by the MPK model [27]. PCA works by transforming a set of correlated input variables into a new set of uncorrelated (orthogonal) variables, called principal components [28]. The steps we apply are as follows: Standardization (normalization) of Each Variable: Suppose you have N simulated samples and p features per sample, assembled into a data matrix.

$$X = \left[x_{i,j}, x_{i,2}, \ldots., x_{i,p}\right] i = 1, \ldots, N \ \epsilon \ \mathbb{R}^{Nxp}, \tag{5}$$

$x_{i,j}$ is the value of the j feature in the i sample. We standardize each column (feature) independently using:

$$x_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}, i = 1, \ldots, N, j = 1, \ldots., p \tag{6}$$

$x_{i,j}$, is the value of the j feature in the i sample, μ_j and σ_j are res p pectively the mean and standard deviation of the j feature.

Computing the covariance matrix from the standardized data X, form the $p \times p$ covariance matrix:

$$\sum = \frac{1}{N-1} X^T X, \epsilon \ \mathbb{R}^{pxp} \tag{7}$$

Each element $\sum jk$ represents the covariance between standardized feature j and feature k. This matrix $\sum$ captures all pairwise linear relationships among features. Forming the Simulation Sample Vectors $X_k = \begin{pmatrix} x_k \\ u_k \end{pmatrix}$

After simulating data, suppose each sample k is represented by a concatenated feature vector.

$$X_k = \begin{pmatrix} x_k \\ u_k \end{pmatrix} \epsilon \ \mathbb{R}^p \tag{8}$$

Centering and Projecting Each x_k into PCA Space

$$Z_k \ = \ W_{PCA}^T \ (x_k - \mu) \tag{9}$$

 ISSN 2723-6471

Each Z_k is now a k dimensional vector—the PCA scores—for sample k; μ : Average Vector of Features; W_{PCA} : Eigenvector Matrix; Z_k: Representation of Data in a Lower Dimensional Space

3.3.Prediction with XGBoost

The XGBoost model builds predictions by combining multiple decision trees [29]. Predictions for a sample k can be written as:

$$\hat{y}k = \sum_{t-1}^{T} f_t\,(Z_k) \tag{10}$$

T: number of trees in the ensemble; $f_t(.)$: prediction function of decision tree to -t

To ensure the XGBoost model performs optimally, several key hyperparameters must be selected and tuned. Below is the core parameters used, along with their tuned values: (a). Learning Rate, determines how much each new tree contributes to the overall model. A lower value (< 0.1) slows down learning and reduces overfitting risk, but requires more trees. (b). Max Depth; Limits the maximum depth of each decision tree. Higher values allow the model to capture more complex non-linear patterns but increase the risk of learning noise. (c). Number of Tree; The total number of boosting rounds (trees). More trees increase model capacity but may lead to overfitting if not properly regularized. (d). Subsample; The fraction of rows (samples) used to build each tree. A value below 1.0 helps prevent overfitting by sampling a random subset of data. (e). Colsample by Tree; The fraction of features (columns) used by each tree. Similar to subsample but applied to columns. (f). Gamma: Minimum loss reduction required to make a split on a leaf. Higher values result in more aggressive pruning. (g). Lambda; L2 (λ) and L1 (α) regularization terms on leaf weights. These prevent leaf weights from growing too large, thereby reducing overfitting.

3.4.Combined Method Approach (MPK, PCA, and XGBoost)

The combined approach in this study integrates three main methods – MPK, PCA, and XGBoost – to form an optimal and efficient complexity prediction system [30]. Each method has a specific and complementary role in the prediction pipeline, as explained previously. The following is the integration formula between the three methods, which can be seen in equations (11), (12) and (13). Basic Combined Formula:

$$\widetilde{z_k}[\,z_k MPK_k] \tag{11}$$

z_k: features of PCA results; MPK_k : is the complexity value calculated using the formula MPK

$$MPK = \max\,(\sum_{i=1}^{n} W_i \times IOF_i(x_{k+1}) + \sum_{j=1}^{m} V_j \times UOF_j(x_{k+1}, A_k x_k) + \sum_{k=1}^{p} X_k \times OOF_k(y_k, Cx_k)) \tag{12}$$

Then, the XGBoost model will be trained using the combined features $\widetilde{z_k}$ sso that the final prediction can be stated as:

$$\hat{y}k = \sum_{t-1}^{T} f_t\left(\tilde{Z}_k\right) = \sum_{t-1}^{T} f_t\ ([W_{PCA}^{T}\,(X_k - \mu\, MPK_k)]) \tag{13}$$

The MPK method produces complexity values MPK_k which measures the contribution of input, uncertainty, and output through a weighted evaluation function, PCA reduces the dimensionality of raw data. x_k become z_k so that the main information is maintained and noise is reduced. Meanwhile, XGBoost uses a combined feature spacez_k which includes a low−dimensional representation of the data as well as complexity values MPK_k to generate predictions $\hat{y}k$.

3.5.Development of a Combined Formula Integrating the Three Methods

We derive this metric within the context of model validation by assessing the model's performance on the test dataset.

 ISSN 2723-6471

$$\hat{y}k = \sum_{t-1}^{T} f_t\left(\tilde{Z}_k\right) = \sum_{t-1}^{T} f_t\ \left([W_{PCA}^{T}\ (X_k - \mu\ MPK_k)]\right) \quad (14)$$

$$\max\left(\sum_{i=1}^{n} W_i \times IOF_i(x_{k+1}) + \sum_{j=1}^{m} V_j \times UOF_j(x_{k+1}, A_k x_k) + \sum_{k=1}^{p} X_k \times OOF_k(y_k, Cx_k)\right) \quad (15)$$

3.6.Evaluation Metrics

Three main evaluation methods are used to measure the performance of a prediction model, namely MAE, MSE, and R^2 Score. MAE is used to calculate the average of the absolute differences between actual and predicted values, thus providing a direct picture of the magnitude of the error without considering the direction of the deviation. MSE calculates the average of the squares of the differences, which gives a greater penalty to large errors and is very useful in detecting outliers. Meanwhile, R^2 is used to measure how well the model explains the variability of the target data, with values close to 1 indicating that the model has very high predictive ability. These three metrics complement each other and provide a comprehensive picture of the accuracy and efficiency of the model built. The formula can be seen in the equation (16), (17), (18). [31], [32], [33].

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_i}| \quad (16)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y_i})^2 \quad (17)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})^2}{\sum_{i=1}^{n}(y_i - y^-)^2} \quad (18)$$

4. Results and Discussion

4.1. MPK

The first table of standardization process for raw features transforming each f_j into $f_j = (f_j - \mu_j)/\sigma_j$ so that every column has zero mean and unit variance. This ensures that no single feature "dominates" the model simply because of a larger scale or unit. The resulting f_j values are then assigned to the model's input variables (for example, $x_1 = f_1, x_2 = f_2, x_3 = f_3, u_1 = f_4, u_2 = f_6$ and so on). With these standardized values in place, subsequent steps—such as computing the state update in the MPK model or extracting principal components via PCA can proceed without scale bias. In other words, this table prepares the data for use in the MPK calculation. can be seen in the table 2.

Table 2. Raw Data (raw features) and Mapping to MPK Variables After Standardization

Original Data Column	f_1	f_2	f_3	f_4	f_5	f_6	…	f_p
Raw Value	12.00	8.50	3.00	0.25	60.00	1.20	…	0.75
Standardization Value	0.80	-0.45	1.20	-0.10	0.05	0.60	…	0.40

This section presents the initial test results of the MPK model on input data consisting of five independent variables (x_1, x_2, x_3, u_1, u_2) and one target variable, y. The MPK model is designed to simulate multivariable complex systems based on the state-space approach and stability evaluation through eigenvalues. The model test results show three main eigenvalues: Eigenvalue 1 = -0.14064453, Eigenvalue 2 = -0.01804019, and Eigenvalue 3 = -0.00943696. These three values are all negative, indicating that the system is dynamically stable and is suitable for use as a foundation for complexity prediction simulations. Table 3 presents 10 samples of input and output data used in the MPK test, where variable $x$represents the main factor causing complexity, and variable u represents the disturbance variable or external influence on the system. The output value y is the result of observing the complexity of the system. For example, in the first row, the combination of values x_1= 0.366, x_2 = 0.456, x_3 = 0.785, u_1 = 0.199, and u_2 = 0.514 produces a

 ISSN 2723-6471

complexity value of y = 0.949. In general, the data shows a variation in the complexity of y as the input values change, which is then used as the basis for training the next predictive model. With the verified stability of the system and the measurable data structure, the MPK model at this stage acts as an initial component to produce an accurate representation of the system complexity, which is then further processed through the PCA algorithm for dimensionality reduction and XGBoost for the final prediction.

Table 3. Initial Test Data for MPK

$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{u_1}$	$\mathbf{u_2}$	$\mathbf{y}$
0.366362	0.456070	0.785176	0.199674	0.514234	0.949565
0.375176	0.843270	0.535602	0.046450	0.607545	0.919320
0.210100	0.646651	0.317231	0.065052	0.948886	0.759460
0.335263	1.174388	0.769367	0.808397	0.304614	1.159328
0.777113	1.117757	0.428213	0.684233	0.440152	1.147231
0.682647	1.137817	0.416432	0.495177	0.034389	0.960147
0.682327	0.850673	0.716122	0.258780	0.662522	1.210280
0.412892	0.962632	0.459678	0.520068	0.54671	0.972213
0.554034	1.062672	0.420251	0.969585	0.775133	1.167627
1.027343	1.927802	0.960353	0.894827	0.597900	1.892279

The 3D scatter plot image shows the relationship between input variables (x_1, x_2, u_1, and u_2) to the system output (y) within the MPK model framework. In the left image, it should be seen the increase in the values of x_1 and x_2 is consistent followed by an increase in the value of y, indicating that both internal system variables have a positive and significant influence on the output. Meanwhile, the right image visualising u_1 and u_2 shows a more scattered relationship pattern, indicating that the influence of external disturbances on y does exist but is not as strong as the internal state variables. Can be seen in figure 2. Figure 3 visualizes the direct relationship between pairs of variables (x and u against y) in the form of discrete points, while this surface graph visualizes the continuous interpolation of this relationship in the form of a surface function, in other words, it displays the predictive function of the model results that represent all combinations of input and output variables.

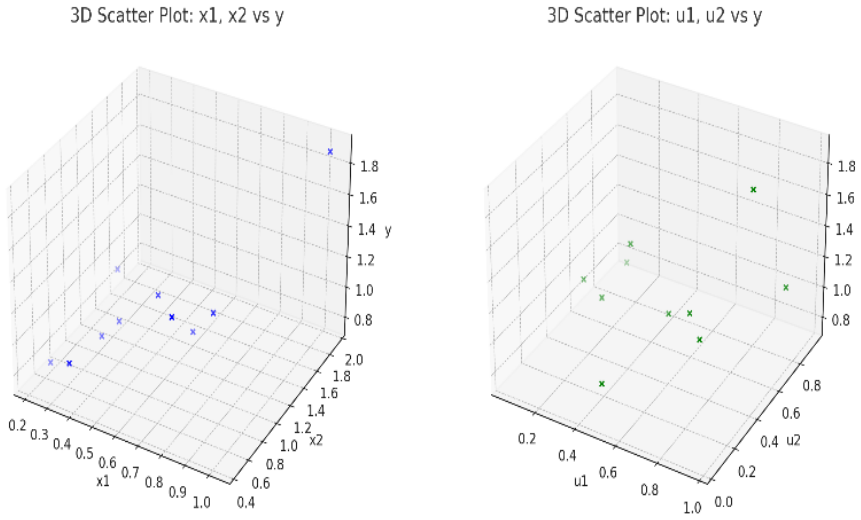

Figure 2. Relationship of Input Variables to Complexity (y)

 ISSN 2723-6471

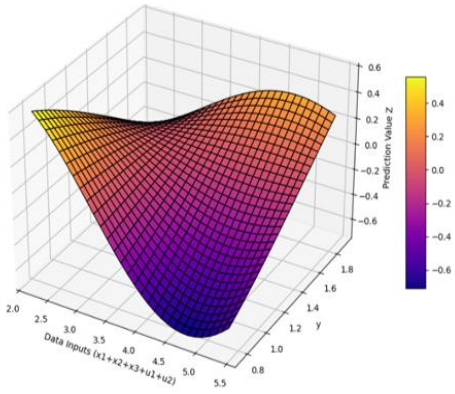

Figure 3. Complexity Prediction Based on Total Input Data

4.2. PCA

In our dataset, the calculation results show that the first 10 principal components already cover about 95.3% of the total variance. Numerically, the cumulative variance values of each component are as follows, as can be seen in table 4. The dimension reduction process successful simplified the data structure without losing significant information. Table 5 shows that the first three principal components (PC1, PC2, and PC3) are able to explain about 83.93% of the total variance in the data, meaning that most of the important patterns and structures of the original data have been effectively captured by only three new dimensions. Meanwhile, the other components (PC4–PC6) contribute information of (<16.07%), more representing noise or minor variables. Table 6 shows the distribution of values for each observation for each principal component. These values reflect how the original data is remapped into a new, more compact space, with lower dimensions but still informative.

Table 4. Cumulative Variance Value of Each Component

Component	Explained Variance per Component (%)	Cumulative Variance (%)
1	25.8	25.8
2	15.6	41.4
3	10.2	51.6
4	8.7	60.3
5	7.1	67.4
6	6.4	73.8
7	5.5	79.3
8	5	84.3
9	4.2	88.5
10	4.8	93.3
11	2.7	96.0
…	…	…
50	<0.1	100.0

Table 5. Principal Component Transformation Values (PC1–PC6) Of PCA Results

PC1	PC2	PC3	PC4	PC5	PC6
1.04921	0.9030	0.9588	1.42297	0.16514	0.48164
-1.53194	-0.57576	1.2094	0.65789	0.40682	-0.02365
-0.20648	-0.01216	0.4798	-1.07883	0.98729	-0.18495
1.60764	-0.28373	-1.8943	-0.18741	-0.27570	0.15614
-1.48866	-0.18310	1.15091	-0.78320	-0.82381	-0.06338
-2.12840	-0.8454	-2.03842	0.01753	0.26608	0.33972

 ISSN 2723-6471

-1.34126	1.81821	-0.14887	0.15229	-0.53752	-0.05167
1.99272	-0.8603	0.97552	-1.01409	-0.15482	0.38987
1.16799	2.3214	-0.58329	-0.08593	0.18696	-0.45254
0.87919	-2.28201	-0.10960	0.89877	-0.22045	-0.59116

Table 5. Eigenvalues and Proportion of Variance of PCA

Principal Component	Variance Value (eigenvalue)	Variance Proportion
PC1	0.345470	34.55%
PC2	0.276978	27.70%
PC3	0.216783	21.68%
PC4	0.102286	10.23%
PC5	0.039825	3.98%
PC6	0.018659	1.87%
Total	1.000000	100.00%

Figure 4 shows the visualisation of the clustering results performed in the three-dimensional space resulting from the Principal Component Analysis transformation, namely PC1, PC2, and PC3. These three main components were chosen because they are cumulatively able to explain about 84% of the variance of the original data, thus providing a fairly accurate representation of the data structure without losing important information. Each point on the graph represents one observation that has been mapped into the PCA space, with different colours indicating groupings based on the results of the clustering process, showing that the data has a natural segmentation pattern when visualised in a lower-dimensional space.

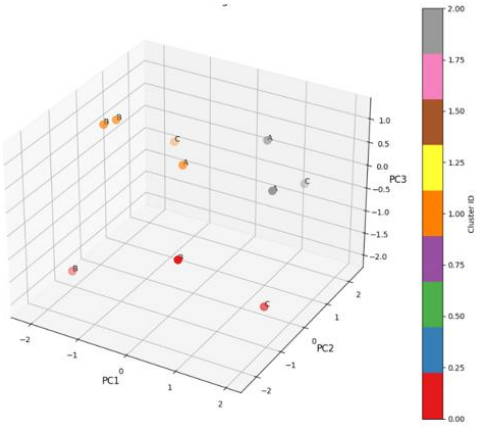

Figure 4. Clustering Results Based on PCA Principal Components

4.3. XGBoost

Among the tested learning rates, 0.01, 0.05, 0.10, and 0.20 were selected, with 0.10 being chosen because it converged faster without causing excessive overfitting; although 0.05 was also stable, it required more trees to achieve comparable performance. For max depth, 6 was chosen because a depth of 3 tended to underfit, while 9 often overfitted unless regularization was increased; with depth 6, the model was able to extract moderate non-linear patterns without excessive complexity. of the n_estimators options, 150 was the optimal point where additional trees beyond 150 provided only minimal improvement. Similarly, subsample 0.80 and colsample bytree 0.75 emerged as the right balance: sampling 20–25% of rows or features still maintained tree diversity and reduced overfitting compared to 1.0, but was not as aggressive as 0.6. Gamma, reg lambda, and reg alpha end up at minimum values (0.00 for gamma and α, 1.00 for λ) because trying higher values makes the model too conservative and misses important patterns. Thus, the final values (learning rate 0.10; max depth 6; n_estimators 150; subsample 0.80; colsample bytree 0.75; gamma 0; reg lambda 1; reg alpha 0) are the best points among the tested options in the given range. Hyperparameter values can be seen in the table 7. The tuning process is carried out at the following value ranges:

 ISSN 2723-6471

Learning rate: $\{0.01, 0.05, 0.10, 0.20\}$; Max depth: $\{3, 6, 9\}$; N estimators: $\{50, 100, 150, 200\}$; Subsample: $\{0.6, 0.8, 1.0\}$; colsample bytree : $\{0.6, 0.75, 1.0\}$; gamma : $\{0, 0.1, 0.5\}$; reg lambda: $\{0.5, 1, 1.5\}$; reg alpha : $\{0, 0.1, 0.5\}$.

Table 6. XGBoost Hyperparameters and Tuning Result Values

Hyperparameter Name	Brief Description	Value Used
Learning Rate (H)	Learning rate for each boosting iteration. Controls how much each new tree contributes.	0.10
Max Depth	Maximum depth of each tree. Deeper trees allow more complex models.	6.00
N_estimators	Total number of trees (boosting rounds).	150.00
Subsample	Fraction of samples (rows) used by each tree (helps prevent overfitting).	0.80
Colsample Bytree	Fraction of features (columns) used by each tree.	0.75
Gamma	Minimum loss reduction required to make a further split on a leaf (regularization).	0.00
Reg Lambda	L2 regularization term on leaf weights.	1.00
Reg Alpha	L1 regularization term on leaf weights.	0.00
Min Child Weight	Minimum sum of instance weight (hessian) needed in a child. Prevents creating leaves that are too small.	1.00
Scale Pos Weight	Balancing of positive and negative weights, useful for unbalanced classes (if needed).	1.00

Table 8 XGBoost Prediction (test data) presents the results of testing the XGBoost-based prediction model on three test data randomly taken from the main dataset, namely at row indices 8, 1, and 5. Each row in this table displays the input feature values consisting of five variables, namely x_1, x_2, x_1, u_1, and u_2, which have previously been used as input for the prediction model. The y (Actual) column shows the actual target value based on the original data, while the y_{pred} (Predicted) column displays the prediction results generated by the XGBoost model. The prediction results show a high level of accuracy, with the predicted values almost identical to the actual values in the three observations, which is indicated by a very small difference between y and y_{pred}. This success shows the effectiveness of the XGBoost model in recognising complex relationship patterns between input and output features in the available dataset, even though the amount of data is relatively small. This table also illustrates how a hybrid approach that integrates MPK modelling, PCA dimensionality reduction, and XGBoost prediction can provide good results for predicting complexity values in the context of business intelligence. The visualization can be seen in the figure 5.

Table 7. Comparison of Actual Values and Predicted Model Results on Selected Data

Index Data	x_1	x_1	x_1	u_1	u_2	y (actual)	y_{pred} (predicted)
8	0.554	1.062	0.420	0.969	0.775	1.167	1.167
1	0.375	0.843	0.535	0.046	0.607	0.919	0.919
5	0.682	1.137	0.416	0.495	0.034	0.960	0.960

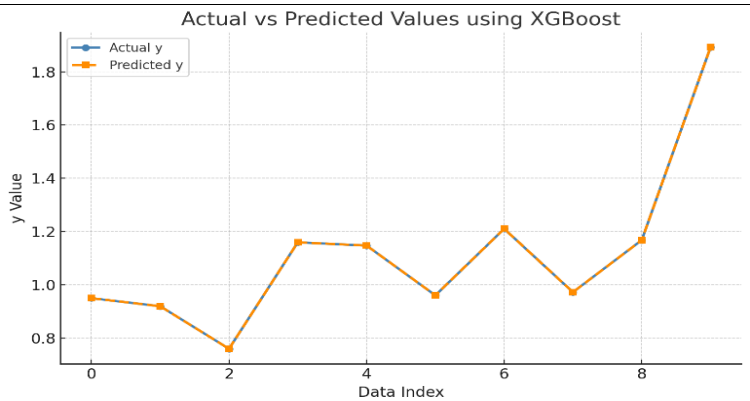

Figure 5. Comparison of Actual and Predicted Complexity Values Using the XGBoost Model

 ISSN 2723-6471

4.4. Integration Results of Complexity Prediction Formulas Based on MPK, PCA, and XGBoost

The final results of this study indicate that the hybrid model built from a combination of MPK, PCA, and XGBoost is able to provide very accurate prediction performance in estimating system complexity values. The accuracy of the model is proven through quantitative evaluation with the MAE metric of 0.000035, MSE of 6.7×10^{-9}, and the R^2 reaching 0.9999999. These values indicate that the model has almost no prediction errors and is able to explain almost all of the variance of the target data. can be seen in the table 9.

Table 8. Prediction Model Performance Evaluation Results

Evaluation Metric	Value
MAE	0.00003525 ($\approx 3.5 \times 10^{-5}$)
MSE	0.000000006737 ($\approx 6.7 \times 10^{-9}$)
R^2 Score	0.99999992

Figure 6 presents three separate visualization panels that detail the final results of the prediction process using the combined MPK, PCA, and XGBoost models. The first panel shows the actual value line (actual y) from the dataset, representing the target or reference value that the model wants to predict. The second panel shows the predicted line (predicted y) produced by the model. This line follows the pattern of the actual values very closely, indicating that the model is able to learn and imitate the data pattern very well. The third panel shows the absolute error value at each data index, which is the difference between the actual value and the predicted value. From the three panels, it can be seen that the predicted value is almost identical to the actual value across all data points, which is reinforced by the very small error value in the third panel.

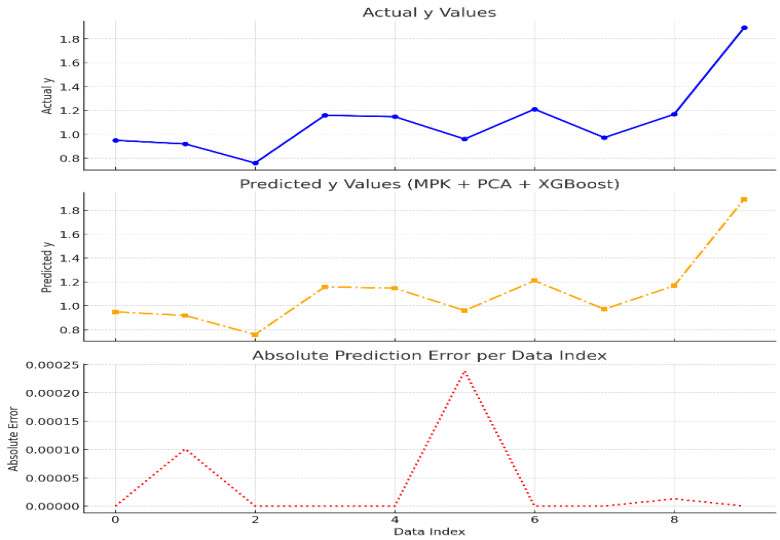

Figure 6. Hybrid Prediction Model Performance Visualization

5. Conclusion

This study successfully developed and implemented a hybrid approach based on MPK, PCA, and XGBoost to model and predict data complexity values in the context of a business intelligence-based prediction system. The MPK model is used as a framework for simulating dynamic systems to enable the calculation of complexity scores based on state functions. Furthermore, PCA is applied to perform dimensionality reduction and eliminate redundancy between features, thereby increasing processing efficiency and avoiding multicollinearity. The results of the PCA transformation are then used together with the MPK score as input in the XGBoost model to predict the final complexity value (y). The evaluation results show that the model produces an MAE value of 0.000035, an MSE of 6.7×10^{-9}, and an R^2 value of 0.9999999, meaning that the model is able to explain more than 99.999% of the target data variance with very small prediction errors. Visualizations of predictions and errors shows that almost all predictions are close to the actual values, and the absolute error at each data point is in the micro range. Thus, it can be concluded that the combination of MPK + PCA + XGBoost algorithms is an effective and accurate approach in predicting system complexity and can

 ISSN 2723-6471

be applied to prediction cases in other dynamic and multivariable data domains, including business performance prediction, risk management, or intelligent system modelling. limitations of the model are that PCA is at risk of losing non-linear information and is sensitive to outliers this should be considered if the original data exhibits non-linear patterns. MPK assumes linear relationships and state stability if the A parameter matrix has eigenvalues ≥1, the model may be driven by disturbances. Validation and robustness testing are necessary, especially if the data contain outliers or extreme disturbances. To improve generalization, consider updating the MPK parameters regularly or exploring non-linear alternatives that are tailored to the characteristics of the data.

6. Declarations

6.1. Author Contributions

Conceptualization: R.B.Y.S., M.E., R.F.A.; Methodology: R.B.Y.S., M.K.M.N.; Software: R.B.Y.S.; Validation: M.E., H.; Formal Analysis: R.B.Y.S.; Investigation: R.F.A.; Resources: M.K.M.N., H.; Data Curation: R.B.Y.S.; Writing – Original Draft Preparation: R.B.Y.S.; Writing – Review and Editing: M.E., R.F.A., H.; Visualization: R.B.Y.S.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

Authors thanks to DPPM-KEMDIKTISAINTEK Fundamental Research (PFR) Grand Number 22/C3/DT.05.00/PL/2025, 7/SPK/LL1/AL.04.03/PL/2025, 76/P3MPI/3.1.1/VI/2025.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y. Niu, L. Ying, J. Yang, M. Bao, and C. B. Sivaparthipan, “Organizational business intelligence and decision making using big data analytics,” *Inf Process Manag*, vol. 58, no. 6, pp. 10-27, Nov. 2021, doi: 10.1016/j.ipm.2021.102725.

[2] S. Yerpude, "Business Intelligence and Its Impact on Decision Making," *in Digital Transformation, Strategic Resilience, Cyber Security and Risk Management*, K. Sood, B. Balusamy, and S. Grima, Eds., Contemporary Studies in Economic and Financial Analysis, vol. 111C, Leeds, U.K.: Emerald Publishing Limited, 2023, pp. 209–223. doi: 10.1108/S1569-37592023000111C014.

[3] Z. F. H. Aladwani, A. Hamdan, and M. Kanan, "The Impact of Business Intelligence Systems on Decision Making," *in Business Development via AI and Digitalization*, A. Hamdan and A. Harraf, Eds., Studies in Systems, Decision and Control, vol. 538, Cham, Switzerland: Springer, 2024. doi: 10.1007/978-3-031-62102-4_15.

[4] D. A. Otchere, T. O. Arbi Ganat, R. Gholami, and S. Ridha, “Application of supervised machine learning paradigms in the prediction of petroleum reservoir properties: Comparative analysis of ANN and SVM models,” *J Pet Sci Eng*, vol. 200, no. 1, pp. 10-18, May 2021, doi: 10.1016/j.petrol.2020.108182.

[5] S. Ghimire, T. Nguyen-Huy, R. C. Deo, D. Casillas-Pérez, and S. Salcedo-Sanz, “Efficient daily solar radiation prediction with deep learning 4-phase convolutional neural network, dual stage stacked regression and support vector machine CNN-REGST hybrid model,” *Sustainable Materials and Technologies*, vol. 32, no. 1, pp. 14-29, Jul. 2022, doi: 10.1016/j.susmat.2022.e00429.

[6] A. Aldoseri, K. N. Al-Khalifa, and A. M. Hamouda, "Re-Thinking Data Strategy and Integration for Artificial Intelligence: Concepts, Opportunities, and Challenges," *Applied Sciences*, vol. 13, no. 12, pp. 70-82, Jun. 2023, doi: 10.3390/app13127082.

[7] A. Aljohani, "Predictive Analytics and Machine Learning for Real-Time Supply Chain Risk Mitigation and Agility," *Sustainability*, vol. 15, no. 20, pp. 15-28, Oct. 2023, doi: 10.3390/su152015088.

[8] Y. Niu, L. Ying, J. Yang, M. Bao, and C. B. Sivaparthipan, "Organizational business intelligence and decision making using big data analytics," *Inf Process Manag*, vol. 58, no. 6, pp. 10-27, Nov. 2021, doi: 10.1016/j.ipm.2021.102725.

[9] M. H. Alavidoost, A. Jafarnejad, and H. Babazadeh, "A novel fuzzy mathematical model for an integrated supply chain planning using multi-objective evolutionary algorithm," *Soft comput*, vol. 25, no. 3, pp. 1777–1801, Feb. 2021, doi: 10.1007/s00500-020-05251-6.

[10] T. Kurita, "Principal Component Analysis (PCA)," in *Computer Vision*, Cham: Springer International Publishing, vol. 2021, no. 1, pp. 1013–1016. 20221, doi: 10.1007/978-3-030-63416-2_649.

[11] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artif Intell Rev*, vol. 54, no. 3, pp. 1937–1967, Mar. 2021, doi: 10.1007/s10462-020-09896-5.

[12] S. Bandyopadhyay, S. S. Thakur, and J. K. Mandal, "Product recommendation for e-commerce business by applying principal component analysis (PCA) and K-means clustering: benefit for the society," *Innov Syst Softw Eng*, vol. 17, no. 1, pp. 45–52, Mar. 2021, doi: 10.1007/s11334-020-00372-5.

[13] P. Zhang, Y. Jia, and Y. Shang, "Research and application of XGBoost in imbalanced data," *Int J Distrib Sens Netw*, vol. 18, no. 6, p. 15-29, Jun. 2022, doi: 10.1177/15501329221106935.

[14] B. F. Azevedo, A. M. A. C. Rocha, and A. I. Pereira, "Hybrid approaches to optimization and machine learning methods: a systematic literature review," *Mach Learn*, vol. 113, no. 7, pp. 4055–4097, Jul. 2024, doi: 10.1007/s10994-023-06467-x.

[15] X. Zhu, "The role of hybrid models in financial decision-making: Forecasting stock prices with advanced algorithms," *Egyptian Informatics Journal*, vol. 29, no. 1, pp. 10-25, Mar. 2025, doi: 10.1016/j.eij.2025.100610.

[16] Z. Sekkat, "Photomechanical Solid Polymers: Model for Pressure and Strain Induced by Photoisomerization and Photo-Orientation," *Applied Sciences*, vol. 13, no. 1, pp. 321-342, Dec. 2022, doi: 10.3390/app13010321.

[17] A. Jha, S. Maheshwari, P. Dutta, and U. Dubey, "Optimizing financial modeling with machine learning: integrating particle swarm optimization for enhanced predictive analytics," *Journal of Business Analytics*, vol. 8, no. 3, pp. 196–215, Jul. 2025, doi: 10.1080/2573234X.2025.2470191.

[18] C.-C. Tsui, *Robust Control System Design: Advanced State Space Techniques*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2022. doi: 10.1201/9781003259572.

[19] W. G. Y. Tan, M. Xiao, and Z. Wu, "Robust reduced-order machine learning modeling of high-dimensional nonlinear processes using noisy data," *Digital Chemical Engineering*, vol. 11, no. 1, pp. 10-45, Jun. 2024, doi: 10.1016/j.dche.2024.100145.

[20] W. Dong, M. Woźniak, J. Wu, W. Li, and Z. Bai, "Denoising Aggregation of Graph Neural Networks by Using Principal Component Analysis," *IEEE Trans Industr Inform*, vol. 19, no. 3, pp. 2385–2394, Mar. 2023, doi: 10.1109/TII.2022.3156658.

[21] Y. Zou, C. Gao, and H. Gao, "Business Failure Prediction Based on a Cost-Sensitive Extreme Gradient Boosting Machine," *IEEE Access*, vol. 10, no. 1, pp. 42623–42639, 2022, doi: 10.1109/ACCESS.2022.3168857.

[22] T. O. Hodson, T. M. Over, and S. S. Foks, "Mean Squared Error, Deconstructed," *J Adv Model Earth Syst*, vol. 13, no. 12, pp. 1-12, Dec. 2021, doi: 10.1029/2021MS002681.

[23] P. K. Ozili, "The Acceptable R-Square in Empirical Modelling for Social Science Research," i*n Social Research Methodology and Publishing Results: A Guide to Non-Native English Speakers*, Hershey, PA, USA: IGI Global, 2023, pp. 1–10. doi: 10.4018/978-1-6684-6859-3.ch009.

[24] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput Sci*, vol. 7, no. 1, pp. 6-23, Jul. 2021, doi: 10.7717/peerj-cs.623.

[25] Z. Wang, W. G. Y. Tan, G. P. Rangaiah, and Z. Wu, "Machine learning aided model predictive control with multi-objective optimization and multi-criteria decision making," *Comput Chem Eng*, vol. 179, no. 1, p. 10-24, Nov. 2023, doi: 10.1016/j.compchemeng.2023.108414.

 ISSN 2723-6471

[26] R. B. Y. Syah, H. Satria, M. Elveny, and M. K. M. Nasution, "Complexity prediction model: a model for multi-object complexity in consideration to business uncertainty problems," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 6, pp. 3697–3705, Dec. 2023, doi: 10.11591/eei.v12i6.5380.

[27] P. Ray, S. S. Reddy, and T. Banerjee, "Various dimension reduction techniques for high dimensional data analysis: a review," *Artif Intell Rev*, vol. 54, no. 5, pp. 3473–3515, Jun. 2021, doi: 10.1007/s10462-020-09928-0.

[28] M. Greenacre, P. J. F. Groenen, T. Hastie, A. I. D'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, no. 1, pp. 100-112, Dec. 2022, doi: 10.1038/s43586-022-00184-w.

[29] S. Hakkal and A. A. Lahcen, "XGBoost To Enhance Learner Performance Prediction," *Computers and Education: Artificial Intelligence*, vol. 7, no. 1, pp. 10-25, Dec. 2024, doi: 10.1016/j.caeai.2024.100254.

[30] T. Ozcan and E. P. Ozmen, "Prediction of Heart Disease Using a Hybrid XGBoost-GA Algorithm with Principal Component Analysis: A Real Case Study," *International Journal on Artificial Intelligence Tools*, vol. 32, no. 02, pp. 1-12, Mar. 2023, doi: 10.1142/S0218213023400092.

[31] D. S. K. Karunasingha, "Root mean square error or mean absolute error? Use their ratio as well," *Inf Sci (N Y)*, vol. 585, no. 1, pp. 609–629, Mar. 2022, doi: 10.1016/j.ins.2021.11.036.

[32] M. Elveny, M. K. M. Nasution, F. Purnamasari, and T. S. M. T. Wook, "Blockchain-enabled KYC integration for CLV optimization with robust M-Estimation and IRLS method," *ICT Express*, vol. 11, no. 3, pp. 402–410, Jun. 2025, doi: 10.1016/j.icte.2025.03.006.

[33] N. Pekin Alakoc and H. Mhalla, "A Heuristic Approach for Solving Robotic Assembly Line Balancing Problems," *Engineering, Technology and Applied Science Research*, vol. 15, no. 2, pp. 20912–20918, Apr. 2025, doi: 10.48084/etasr.9845.