The Aeronautical Journal (2024), 1–20
doi:10.1017/aer.2024.143

ROYAL AERONAUTICAL SOCIETY

RESEARCH ARTICLE

Method of fault prediction for avionics components based on stacking regression

W.H. Li[1], G. Li[1,2], Y. Liu[1], J.T. Ma[1], Z.D. Wu[1], X. Tang[1], W.C. Sun[1] and T.Z. Wen[1]

[1]Aviation Combat Service Academy, Naval Aviation University, Yantai, China
[2]Repair Brigade, 77120 Unit of the Chinese People's Liberation Army, Chengdu, China
Corresponding author: G. Li; Email: 504123921@qq.com

Received: 16 July 2024; **Revised:** 19 September 2024; **Accepted:** 14 November 2024

Keywords: multi-step prediction; avionics components; meta-regressor; stacking regression; fault prediction

Abstract

As avionics systems become increasingly complex, traditional fault prediction methods are no longer sufficient to meet modern demands. This paper introduces four advanced fault prediction methods for avionics components, utilising a multi-step prediction strategy combined with a stacking regressor. By selecting various standard regression models as base regressors, these base regressors are first trained on the original data, and their predictions are subsequently used as input features for training a meta-regressor. Additionally, the Tree-structured Parzen Estimator (TPE) algorithm is employed for hyperparameter optimisation. The experimental results demonstrate that the proposed stacking regression methods exhibit superior accuracy in fault prediction compared to traditional single-model approaches.

Nomenclature

EI	expected improvement
ESN	echo state networks
FI	fault indicator
GenAI	generative artificial intelligence
GRU	gate recurrent unit
IMU	inertial measurement unit
MAE	mean absolute error
MAPE	mean absolute percentage error
MDBN	multi-deep belief networks
PHM	Prognostics and Health Management
TPE	Tree-structured Parzen Estimator
RMSE	root mean square error
RUL	remaining useful life
SINS	strapdown inertial navigation system
SVM	support vector machines
SVR	support vector regression

Symbols

b	bias term
C	regularization parameter
w	weight vector
λ	regularization parameter

1.0 Introduction

Electronic equipment is a crucial component of aviation systems, with its performance directly impacting the operational state of such equipment [1]. As modern electronic technology continues to advance, the internal complexity of electronic devices has increased, necessitating enhanced maintenance and support. Prognostics and Health Management (PHM) technology has been thoroughly researched and widely applied in both military and civilian sectors, particularly in supporting aviation equipment [2].

PHM technology utilises various sensors to collect data and employs intelligent algorithms to diagnose and predict the health status of systems or components, thereby facilitating informed maintenance and health management decisions. This ensures the stable and reliable operation of the systems or components involved [3]. Fault prediction, a key component of PHM, primarily focuses on predicting the potential failure times of systems or components [4]. Fault prediction techniques are broadly categorised into two types: model-based and data-driven. Equipment failure is fundamentally influenced by various physical stressors. Model-based approaches, which utilise physical models to describe these phenomena, can provide more explainable and potentially more reliable predictions. These models can capture the underlying physical processes leading to failure, offering insights that purely data-driven methods might miss. Despite these advantages, model-based approaches also have limitations. They require detailed knowledge of the system's physical properties and failure mechanisms, which can be difficult to obtain for complex systems. Additionally, these models may not generalise well to different operating conditions or new types of failures. In contrast, data-driven methods, such as the stacking regression approach proposed in this paper, can leverage large amounts of historical data to identify patterns and make predictions without needing detailed physical models. This makes them more flexible and easier to apply to a wide range of systems and conditions. Model-based methods depend on the physical failure models specific to each object, which follow distinct failure evolution laws. In contrast, data-driven fault prediction methods rely solely on historical data analysis and the use of advanced intelligent algorithms to develop prediction models, without the need for prior system knowledge [5]. While model-based methods require extensive physical domain expertise and often suffer from limited generalisation capabilities, their effectiveness diminishes with increasing system complexity. On the other hand, data-driven methods, bolstered by advancements in state detection technology and an abundance of test data, have demonstrated strong adaptability and are gaining prominence among researchers in the field of fault prediction [6].

The primary focus of fault prediction research is often on time series data, which involves analysing historical sequences to forecast future trends. The ability to effectively recall, extract and utilise historical data is crucial for the success of time series prediction. Machine learning methods, known for their speed and independence from prior knowledge, have been extensively applied in avionics fault prediction. Liang [7] introduced a fault prediction approach for avionics products using a fusion of multi-deep belief networks (MDBN), which addresses prediction biases caused by distribution differences between target and historical data through transfer training of multiple MDBN models, thereby enhancing prediction accuracy. Experimental results validated the effectiveness of this method. Zhang [8] developed a fault prediction method for the instrument landing system using the Gate Recurrent Unit (GRU), focusing on the course beacon and employing monitoring parameters as fault indicators to calculate future fault probabilities based on the membership function of these parameters. Mitici [9] applied various regression prediction methods to estimate the health status and remaining life of batteries in electric vertical takeoff and landing aircraft, with experimental results supporting the feasibility of these methods. Gao [10] proposed a long-term fault prediction method for avionics using echo state networks (ESN), and tested its performance on various avionics datasets. While existing data-driven fault prediction methods for avionics primarily rely on single models, stacking regression – an ensemble learning technique – effectively combines multiple base regression models through a meta-regressor, often yielding superior predictive performance compared to single models [11]. This approach has been successfully applied

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

in various fields such as stock [12], weather [13] and photovoltaic power [14] prediction, yet its use in avionics fault prediction remains limited.

To address the limitations of traditional single models and enhance prediction accuracy in avionics fault prediction, this paper introduces four multi-step prediction methods based on stacking regression. Each method employs one of four typical regression techniques – support vector regression (SVR), ridge regression, lasso regression, and elastic_net regression – as the meta-regressor, with the remaining three serving as base regressors. Initially, multiple base regression models independently predict data, which are then fed into the meta-regressor in the second layer. This process synthesises the predictive strengths of each base model. Hyperparameters are optimised using the TPE algorithm to further enhance prediction accuracy and model generalisation. The effectiveness and advancement of these methods were validated through experiments conducted on simulated circuit components and inertial measurement unit components.

It is worth noting that while generative artificial intelligence (GenAI) has gained attention for its ability to generate new data points and simulate complex data distributions, the fault prediction method proposed in this paper is based on stacking regression, a form of ensemble learning. Unlike GenAI, which focuses on data generation, stacking regression aims to improve predictive accuracy by combining multiple regression models. This distinction is important as it highlights the specific approach and advantages of using stacking regression for fault prediction in avionics components.

2.0 Typical regression prediction models

2.1 Support vector regression

SVR [15] is a significant subset of support vector machines (SVM). The fundamental concept of the SVR algorithm involves identifying a regression plane that minimises the distance to all data points in the dataset.

For linear kernel SVR, the model is represented as follows:

$$f(x) = w^T x + b \tag{1}$$

where w is the weight vector, x is the input feature vector, and b is the bias term.

The objective of SVR is to identify a function $f(x)$ such that the discrepancy between the actual target values y of most data points x and the predicted values $f(x)$ remains within a predetermined threshold ϵ, while simultaneously minimising the model's complexity. This goal is accomplished by solving the following optimisation problem:

$$\min_{w,b} \left\{ \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) \right\} \tag{2}$$

C is the regularisation parameter, which balances the trade-off between the error term and model complexity.

The minimisation function is subject to the following constraints:

$$\begin{cases} y_i - (w^T x_i + b) \le \varepsilon + \xi_i, \\ (w^T x_i + b) - y_i \le \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \ge 0 \end{cases} \tag{3}$$

In the equation, ξ_i and ξ_i^* represent slack variables. The SVR regression function is derived by solving the optimisation problem:

$$f(x) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) x_i^T x + b \tag{4}$$

In the equation, α_i and α_i^* act as Lagrange multipliers.

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

2.2 Ridge regression

Ridge regression [16] is a linear regression technique designed to address multicollinearity in data by incorporating an L2 regularisation term. This addition enhances both the stability and predictive capability of the model. The objective of ridge regression is to minimise the following cost function:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \tag{5}$$

In this equation, $h_\theta(x)$ represents the predicted value of the model. For a linear model, $h_\theta(x) = \theta^T x$, where $x^{(i)}$ is the feature vector of the i-th observation, and $y^{(i)}$ is the corresponding target variable. The term θ denotes the model parameters, which include both the intercept and slope. λ is the regularisation parameter that controls the strength of the regularisation. The number of samples is represented by m, and n denotes the number of features. The first term of the equation is the mean squared error, while the second term, the regularisation term, is the L2 norm.

2.3 Lasso regression

Lasso regression [17] is a variant of linear regression that combats overfitting and facilitates automatic feature selection by incorporating an L1 regularisation term.

The cost function for lasso regression is as follows:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \left|\theta_j\right| \tag{6}$$

In this equation, $h_\theta(x)$ represents the predicted value of the model. For a linear model, $h_\theta(x) = \theta^T x$, where $x^{(i)}$ is the feature vector of the i-th observation, and $y^{(i)}$ is the corresponding target variable. The term θ denotes the model parameters, which include both the intercept and slope. λ is the regularisation parameter that controls the strength of the regularisation. The number of samples is represented by m, and n denotes the number of features. The first term of the equation is the mean squared error, while the second term, the regularisation term, is the L1 norm.

2.4 Elastic_net regression

Elastic_net regression [18] is a linear regression model that merges the features of lasso regression and ridge regression by simultaneously incorporating both L1 and L2 regularisation terms for parameter estimation.

The cost function for elastic_net regression is as follows:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \left(\alpha \sum_{j=1}^{n} \left|\theta_j\right| + \frac{1-\alpha}{2} \sum_{j=1}^{n} \theta_j^2\right) \tag{7}$$

In this equation, $h_\theta(x)$ represents the predicted value of the model. for a linear model, $h_\theta(x) = \theta^T x$, where $x^{(i)}$ denotes the feature vectors of the i-th observation, and $y^{(i)}$ is the corresponding target variable. The term θ signifies the model parameters, including both the intercept and slope. λ is the regularisation parameter that controls the overall strength of regularisation. α is a parameter ranging between 0 and 1, used to balance the contributions of L1 and L2 regularisation. m represents the number of samples and n is the number of features. The first term of the equation is half of the mean squared error, while the second term, the regularisation term, encompasses both the L1 norm, which encourages sparsity, and the L2 norm, which ensures smoothness of the parameter values.

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

3.0 Stacking regression prediction models

3.1 Multi-step stacking prediction

Ensemble learning is a sophisticated machine learning technique that has garnered considerable attention and achieved notable success in both academic and industrial settings. This method involves training multiple individual models and combining their outputs. Stacking ensemble, a key approach within ensemble learning, primarily focuses on integrating the predictions from multiple base models by constructing a meta-learner.

Stacking regression employs a two-layer structure where the first layer consists of multiple base regressors that directly learn from the original training set. The second layer, or the meta-regressor, uses the predictions from these base regressors as input features for training. This approach allows the meta-regressor to amalgamate the predictive strengths of each base regressor, enhancing overall effectiveness through several key aspects:

1. Model diversity: Stacking regression utilises a variety of base regression models, each differing in their response to various data distributions and problem types. By integrating these diverse models, stacking regression leverages the unique strengths of each, thereby mitigating the risks associated with dependency on a single model.
2. Weight adjustment: The meta-regressor's primary role is to determine the most effective way to combine the base models' predictions. It assigns greater weights to models that perform better under specific conditions, optimising the final prediction outcome.
3. Reduction of overfitting: Single models may tend to overfit the training data. Stacking regression addresses this by blending predictions from multiple models, which helps in reducing overfitting. The meta-regressor plays a crucial role in identifying overfitting tendencies among the base models and adjusts their influence accordingly, thus enhancing the robustness of predictions on new, unseen data.

Using lasso regression and ridge regression with elastic_net regression as base regressors may introduce correlated updates to the meta-regressor, which from an information theory perspective, could be seen as suboptimal. However, our choice is driven by the complementary strengths of these models in handling different aspects of the data. Lasso regression is effective for feature selection and sparsity, ridge regression addresses multicollinearity, and elastic_net combines both L1 and L2 regularisation to balance these effects. To mitigate potential correlation issues, we ensure that the meta-regressor is trained on the residuals of the base regressors' predictions, which helps in capturing the unique contributions of each base model and reducing redundancy.

Stacking regression significantly improves prediction accuracy and model generalisation by harnessing the collective capabilities of multiple models and fine-tuning their integration through a sophisticated meta-regressor. This paper employs computationally efficient typical regression models such as support vector regression, ridge regression, lasso regression, and elastic_net regression as base predictors. One of these models is designated as the meta-regressor, with the remaining serving as base regressors, leading to the proposal of four distinct stacking regression methods, as detailed in Table 1. Additionally, this paper incorporates lag features in time series prediction, utilising historical data values as input features to forecast future trends, with the number of lag features tailored to meet practical requirements.

Taking the proposed stacking-elastic_net regression prediction model as an example, as illustrated in Fig. 1, the structure of stacking regression is depicted. The first layer comprises SVR, ridge regression and lasso regression as base regressors. These base regressors independently process the training data and forward their prediction results as input features to the second layer's meta-regressor, elastic_net. The meta-regressor, elastic_net, integrates the predictions from the first layer and employs multi-step recursive prediction to generate final forecasts for the desired prediction time steps. This stacking regression method amalgamates the prediction results from multiple base regressors to bolster the overall performance of the model. The final output, P_{final}, represents a comprehensive prediction of the test dataset.

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

Table 1. Overview of proposed stacking regression methods

Model name	Base regressors	Meta-regressor
Stacking-SVR	Ridge, Lasso, Elastic_Net	SVR
Stacking-ridge	SVR, Lasso, Elastic_Net	Ridge
Stacking-lasso	SVR, Ridge, Elastic_Net	Lasso
Stacking-elastic_net	SVR, Ridge, Lasso	Elastic_Net

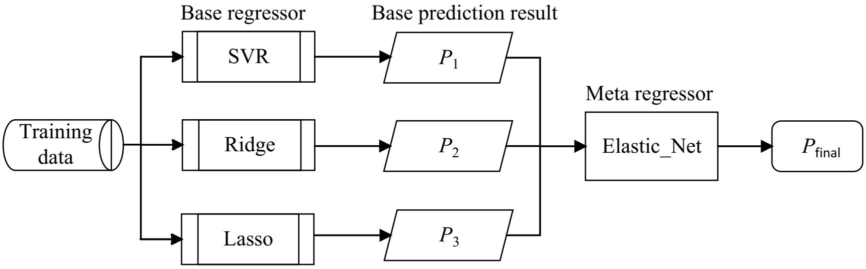

***Figure 1.** Structure of the stacking-elastic_net regression model.*

Algorithm 1: Stacking-Elastic_net

```
Input:  Training data (X_train, y_train), Test data (X_test, y_test)
Output: Final prediction results for test data

Initialize base regressors: SVR, Ridge, Lasso
Initialize meta-regressor: Elastic_Net
// Train base regressors
For each base_regressor in [SVR, Ridge, Lasso]:
    base_regressor.fit(X_train, y_train)
End for
// Generate meta-features for training meta-regressor
Create an empty matrix X_meta_train
For each base_regressor in [SVR, Ridge, Lasso]:
    predictions = base_regressor.predict(X_train)
    Append predictions as a new column to X_meta_train
End for
// Train meta-regressor
Elastic_Net.fit(X_meta_train, y_train)
// Generate meta-features for final prediction
Create an empty matrix X_meta_test
For each base_regressor in [SVR, Ridge, Lasso]:
    predictions = base_regressor.predict(X_test)
    Append predictions as a new column to X_meta_test
End for
// Make final predictions using meta-regressor
final_predictions = Elastic_Net.predict(X_meta_test)
// Return final prediction results
Return final_predictions
```

The pseudocode for the stacking-elastic_net regression model is outlined as Algorithm 1.

3.2 Hyperparameter optimisation with TPE

In 2011, James [19] from Harvard University proposed the TPE, which uses a tree structure to represent the relationships between hyperparameters and can effectively solve multi-dimensional optimisation problems. Additionally, it requires fewer iterations to find satisfactory hyperparameters. Therefore, TPE is an ideal method for hyperparameter tuning in stacking regression models. This method can adapt to different objective functions and improve the quality of hyperparameter selection.

The TPE algorithm defines $p(x|y)$ using two density functions as follows:

$$p(x|y) = \begin{cases} l(x) & \text{if} \quad y < y^* \\ g(x) & \text{if} \quad y \geqslant y^* \end{cases} \tag{8}$$

Here, x represents the observation point, which is the hyperparameter vector of the model to be optimised; y is the observation value, the outcome of the objective function (loss or evaluation function) for the given parameter x; and y^* is a threshold, representing a specific quantile of the TPE algorithm used to divide the observations into two density functions, $l(x)$ and $g(x)$, with the sum ranging from 0 to 1. The algorithm sets y^* based on existing observations, segregating them into two distinct density functions: $l(x)$ is formed by those observations $\left\{x_{(i)}\right\}$ whose loss is $f(x_{(i)}) < y^*$, and $g(x)$ consists of the remaining observations. TPE employs Bayesian optimisation principles to minimise ineffective search space.

The expected improvement (EI) strategy employed by the TPE algorithm generates new observation points aimed at maximising EI. The Bayesian approach optimises the EI acquisition function, letting $p(y)p(x\ |y\ )$ be $p(x, y)$, thus defining EI as follows:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y)p(y|x)dy = \int_{-\infty}^{y^*} (y^* - y)\frac{p(x|y)p(y)}{p(x)}dy \tag{9}$$

Let $\gamma = p(y < y^*)$, to simplify the above formula, construct the denominator as: $p(x) = \int p(x|y)p(y)dy = \gamma l(x) + (1 - \gamma)g(x)$.

Secondly, for the molecule, we can get the formula:

$$\int_{-\infty}^{y^*} (y^* - y)p(x|y)p(y)dy = l(x)\int_{-\infty}^{y^*} (y^* - y)p(y)dy = \gamma y^* l(x) - l(x)\int_{-\infty}^{y^*} p(y)dy \tag{10}$$

Consequently, EI can be simplified to:

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x)\int_{-\infty}^{y^*} p(y)dy}{\gamma l(x) + (1 - \gamma)g(x)} \propto \left(\gamma + \frac{g(x)}{l(x)}(1 - \gamma)\right)^{-1} \tag{11}$$

From this formula, maximising EI involves ensuring the probability of $l(x)$ is high and that of $g(x)$ is low. In each iteration, the algorithm returns the candidate hyperparameter x^* with the highest EI value, $x^* = \arg\max EI_{y^*}$. During the maximisation process, the hyperparameter x that achieves the highest $l(x)$ and the lowest $g(x)$ probabilities obtains the highest EI value. The TPE algorithm uses $l(x)$ and $g(x)$ to generate a collection of hyperparameter samples and evaluates x by the ratio of $l(x)/g(x)$. In the hyperparameter optimization of the Stacking regression model, the new hyperparameter x is used to adjust the parameters of the Stacking regression model, training is conducted to obtain the observed value y, then the new sample point observations are compared with the original observations to update the probability model, thereby selecting the optimal result corresponding to the hyperparameter x

To address concerns about diluting a good solution with a poor one and to provide a probabilistic measure of confidence, we incorporate Bayesian optimisation principles within the TPE framework. This allows us to quantify the uncertainty associated with each hyperparameter configuration. By evaluating the posterior distribution of the hyperparameters, we can obtain a probabilistic measure of confidence for each configuration, ensuring that the selected hyperparameters are not only optimal but also robust.

This paper employs time series five-fold cross-validation to optimise hyperparameters, as illustrated in Fig. 2, where the time series is sequentially divided into six data blocks. In the first fold of validation, after training on the first data block, the subsequent second data block is used for validation. In the second fold, the combined first and second data blocks serve as the training set, with the third data block used for validation. This sequence continues until the final, sixth data block is also utilised as a validation set. Each fold of the time series cross-validation uses different combinations of training and

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

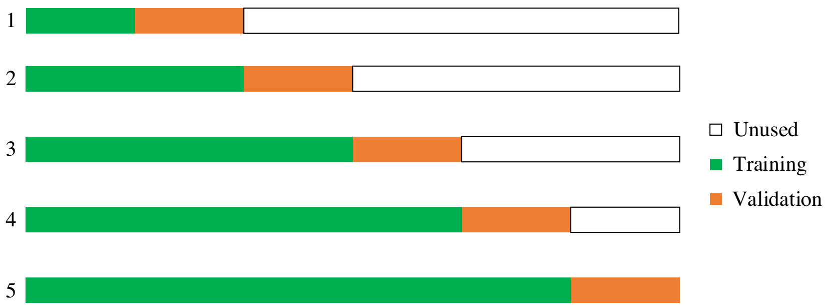

Figure 2. *Time series five-fold cross-validation.*

validation sets, maintaining the sequential integrity of the time series and ensuring data completeness. This method is particularly well-suited for time series data as it allows the model to capture the temporal dynamics without the risk of future information leakage.

3.3 Prediction evaluation metrics

The prediction accuracy metrics discussed in the text include root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The formulas for these metrics are as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|\hat{y}(i) - y(i)|^2} \tag{12}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}(i) - y(i)\right| \tag{13}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}(i) - y(i)}{y(i)}\right| \tag{14}$$

In these formulas, $y(i)$ represents the actual value, $\hat{y}(i)$ is the predicted value, and n is the number of samples. To ensure consistency with the RMSE and MAE metrics, the MAPE metric is expressed in decimal form.

The implementation framework for the stacking regression prediction model is depicted in Fig. 3.

4.0 Experimental analysis

In this paper, ridge, lasso, SVR, and elastic_net are utilised as base predictors, and four types of stacking regression methods are proposed: stacking-ridge, stacking-lasso, stacking-svr, and stacking-elastic_net. Each method employs one of the base predictors as the meta-regressor, with the remaining predictors serving as the first-layer base regressors. The proposed stacking methods initially conduct fault prediction experiments on key components of analog circuits, followed by experiments on inertial measurement unit components. These methods are then experimentally compared with single base predictors such as ridge, lasso, elastic_net, and SVR. Throughout the experiments, all methods utilise the TPE optimisation algorithm to determine the optimal hyperparameters. To collect failure data for these rare events, we utilised a combination of historical data from operational records, controlled experiments designed to simulate real-world fault scenarios, and simulation techniques to generate synthetic data.

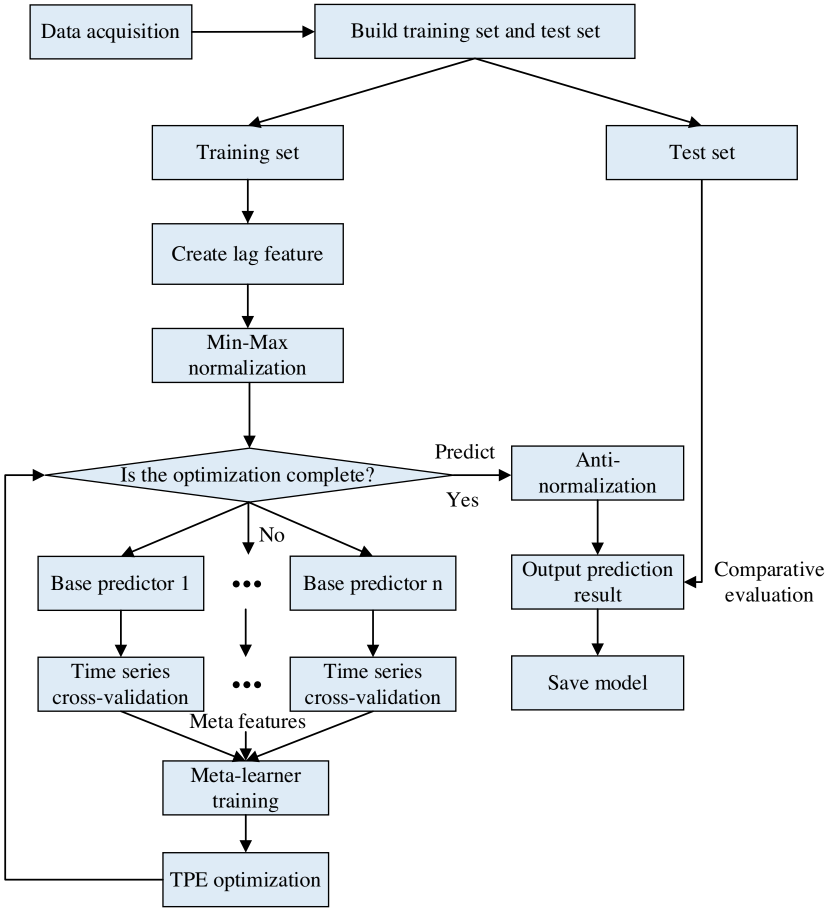

Figure 3. *Implementation framework of the stacking regression prediction model.*

This comprehensive approach ensures that our dataset captures a wide range of potential failure modes, providing a robust basis for developing and validating our fault prediction models.

The training dataset for our study is derived from simulated fault scenarios and historical data collected from extensive testing and operational records of avionics components. This includes both nominal and fault conditions, allowing us to create a robust dataset that captures a wide range of potential failure modes. While zero-shot or anomaly detection models are valuable for identifying rare events, our focus on stacking regression is driven by the need for precise fault prediction and the ability to leverage multiple regression models to improve predictive accuracy.

When the health of a component is calculated, the remaining useful life (RUL) or prognostics is then the time it takes to go from the estimated health state to a predefined failure threshold. Accurate diagnostics are essential for effective prognostics, as they provide the necessary information to estimate the current health state of the component. In our experiments, we focus on enhancing the accuracy of fault prediction, which is a critical aspect of both diagnostics and prognostics. The TPE optimisation algorithm, part of the Optuna [20] hyperparameter tuning framework, was introduced in 2019 by Takuya Akiba and colleagues from PFN, Japan. To enhance computational efficiency, the range of hyperparameters explored by TPE is discretised, with the specific parameter value ranges detailed in Table 2. Additionally, ridge, lasso, SVR, and elastic_net are all implemented using the scikit-learn library, with the SVR method employing a linear kernel.

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

***Table 2.** Hyperparameter selection range*

Parameter	Parameter meaning	Parameter value range
ridge_alpha	Ridge regression regularisation strength	1×10^{-6}, 1×10^{-5}, 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}
lasso_alpha	Lasso regression regularisation strength	1×10^{-6}, 1×10^{-5}, 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}
svr_c	SVR regularisation strength	1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}, 1×10^{0}, 1×10^{1}
svc_epsilon	SVR insensitivity loss	1×10^{-6}, 1×10^{-5}, 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}
elastic_net_alpha	Elastic_net regression regularisation strength	1×10^{-6}, 1×10^{-5}, 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}
elastic_net_l1_ratio	Elastic_net regularisation L1 to L2 ratio	1×10^{-2}, 5×10^{-2}, 1×10^{-1}, 3×10^{-1}, 5×10^{-1}, 7×10^{-1}, 9×10^{-1}

While our study primarily focuses on data-driven methods, it is important to acknowledge the potential benefits of model-based approaches. Model-based methods utilise physical models to describe the underlying failure mechanisms of components, which can provide more explainable and potentially more reliable predictions. These models can capture the physical processes leading to failure, offering insights that purely data-driven methods might miss. However, model-based approaches require detailed knowledge of the system's physical properties and failure mechanisms, which can be difficult to obtain for complex systems. Additionally, these models may not generalise well to different operating conditions or new types of failures. In our simulations, we have the necessary data and conditions to implement model-based approaches. However, we chose to focus on data-driven methods due to their flexibility and ability to leverage large amounts of historical data without requiring detailed physical models. This makes them more adaptable to a wide range of systems and conditions. To validate our data-driven methods, we conduct extensive simulations that mimic real-world fault scenarios, ensuring that our models are robust and reliable.

In the analog circuit, both the anomaly threshold and the failure threshold are calculated. We first estimate the distribution of the fault indicator (FI) and use the fitted probability distribution model to determine the thresholds, ensuring statistical robustness. For the inertial measurement unit (IMU), the thresholds are based on actual operational experience. For example, a gyroscope drift coefficient exceeding 0.36°/h is considered beyond performance criteria and is set as the fault threshold. This approach combines statistical calculations and practical experience to ensure the reliability and applicability of the threshold selection.

The experiments were conducted in the following environment: an Intel Core i9-13900HX processor with a base frequency of 2.20GHz and a max turbo frequency of 5.60GHz, and 32GB of memory. The software environment included a Windows 11 operating system and the PyCharm 2023.6 integrated development environment, using Python version 3.8.5.

4.1 Fault prediction for key components in analog circuits

Analog circuits are extensively utilised in aviation equipment and play a crucial role in avionics. Failures in these circuits often stem from key, sensitive components. This paper delves into the degradation of circuit performance, particularly focusing on failures induced by the gradual deviation of key component parameters from their normal values. Utilising the circuit fault analysis tool developed in Ref. (21), this study conducted a simulation analysis on circuit performance degradation and gathered characteristic data on component degradation. The research specifically examines a commonly used low-pass elliptical filter circuit, employing stacked regression methods to predict component faults. The structure of the

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

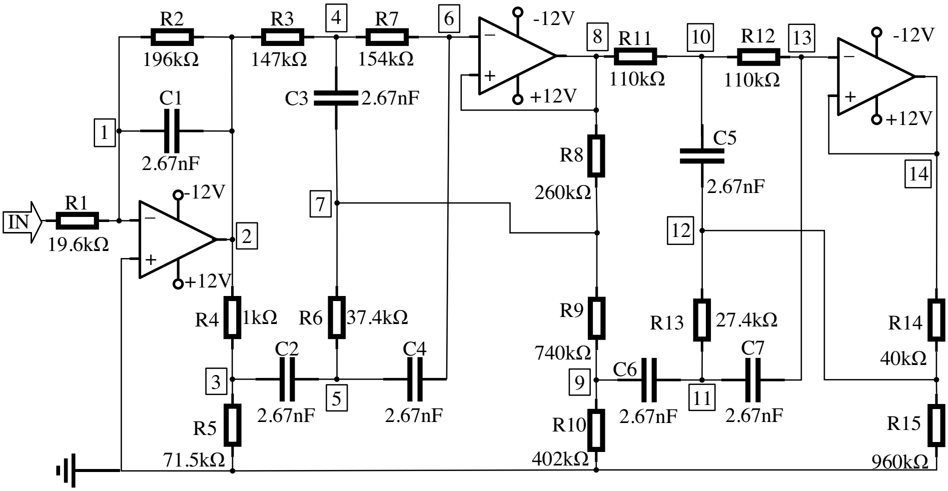

Figure 4. *Schematic diagram of the low-pass elliptical filter circuit.*

low-pass elliptical filter circuit is depicted in Fig. 4. During the experiments, steady-state voltage peaks at 14 measurement points within the circuit were selected as monitoring features to facilitate online monitoring of the circuit's overall operational state.

In real-world applications, the cost of analog interfaces required to measure 14 parameters might exceed the cost of the circuit itself, as depicted in Fig. 4. This raises concerns about the return on investment, especially if the measurements are not gross level data such as input voltage/current and temperature. To address this, we propose a cost-effective approach by focusing on a subset of critical parameters that can still provide sufficient information for accurate fault prediction. By employing feature selection techniques, we can identify the most informative parameters, thereby reducing the number of required measurements and associated costs. This approach ensures a balance between measurement precision and economic feasibility, making the fault prediction method more practical for real-world applications.

A single monitoring feature often falls short in providing a comprehensive and accurate depiction of component performance degradation. Employing multiple features requires complex data fusion calculations. Consequently, this paper embraces the concept of the FI, as proposed in Ref. (22). FI is a state variable formulated based on circuit monitoring features to reflect the overall performance degradation of the circuit. Typically, when component parameters deviate, the FI value exhibits a monotonic trend of change. The methodology for constructing the FI will be discussed in the following section.

Let V_{ij} represent the steady-state voltage peak at node j at the i-th sampling point, and V_j^{normal} represent the steady-state voltage peak of node j under normal conditions. When a parameter of a component in the circuit changes, the deviation of node j's monitoring feature from the normal state can be represented as $D_{ij} = \left|V_{ij} - V_j^{normal}\right|$. Given that the monitoring features (i.e. steady-state voltage peaks) of different nodes vary in their degree of change during component performance degradation, node normalisation is performed on D_{ij}:

$$\overline{D}_{ij} = \frac{D_{ij} - (D_{*j})_{\min}}{(D_{*j})_{\max} - (D_{*j})_{\min}}, \quad i = 1, 2, \cdots, n; \quad j = 1, 2, \cdots, M \tag{15}$$

In the formula, $(D_{*j})_{\max}$ and $(D_{*j})_{\min}$, respectively, represent the maximum and minimum values of node j across all sampling points; n is the number of sampling points; M is the number of monitoring nodes. According to the coarse feature extraction method described in Ref. (23), M monitoring features

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

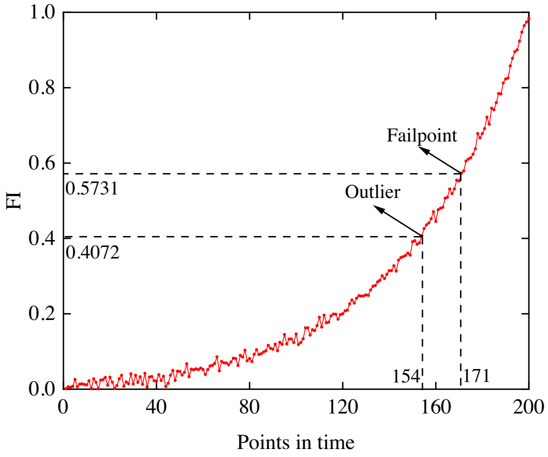

Figure 5. *Trend of FI value changes under R2↓.*

(nodes) are selected, and the FI is defined as:

$$FI_i = \sum_{j=1}^{M} \left\{ \left[\frac{dis(j)}{\sum_{t=1}^{M} dis(t)} \right] \overline{D}_{ij} \right\}, \quad i = 1, 2, \cdots, n \tag{16}$$

In the formula, $dis(j)$ is defined in Ref. (23) as the feature discriminative power of each monitoring feature j. Based on the definition of feature discriminative power, the discriminative power of each monitoring feature is integrated as weight into the construction of the FI, allowing this fault indicator to more accurately reflect the overall working state of the system. In this case, using the coarse feature extraction method, 6 measurement points {4, 5, 6, 9, 10, 11} were selected as key monitoring points, and the degradation process under the reduction of the R2 parameter in the low-pass elliptical filter circuit (denoted as R2↓) was analysed. In the experiment, the degrading component parameter starts from its nominal value Θ, decaying exponentially over time, with the maximum change set to 0.5Θ, and a sampling length of 200. The parameters of non-degrading components vary randomly according to a normal distribution $N(\Theta, (t\Theta/3)^2)$ (where t is a 10% relative tolerance), and the circuit input receives a 3V, 1kHz sine wave signal. Using the defined FI construction method, the changes in FI corresponding to R2↓ can be obtained, as shown in Fig. 5. Due to the influence of the tolerance of non-degrading components, the FI degradation curve is not smooth, which may adversely affect fault prediction.

The distribution of the FI is crucial for setting appropriate thresholds for anomaly detection. From a zero-shot or anomaly detection perspective, if the distribution of FI is known, the threshold can be determined using the inverse cumulative distribution function for a given probability of false alarm. This approach allows us to set a threshold that minimises the likelihood of false alarms while ensuring that true anomalies are detected. In our experiments, we estimate the distribution of FI using historical data and fit it to a suitable probability distribution model. The anomaly threshold is then set as the value corresponding to the 1 - probability of false alarm quantile of this distribution. This method provides a probabilistic measure of confidence in the threshold setting, ensuring robust anomaly detection.

To determine the appropriate distribution model for FI, we perform goodness-of-fit tests, such as the Kolmogorov-Smirnov test, Anderson-Darling test, and Chi-square test, to compare the empirical distribution of FI with various theoretical distributions, including Gaussian, Rayleigh, and others. Preliminary analysis suggests that FI may exhibit a tailed distribution, such as Rayleigh, rather than a Gaussian distribution. The anomaly threshold is then set as the value corresponding to the 1- probability of false alarm

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

Table 3. *Comparative performance metrics of different models*

Model category	Model	RMSE	MAE	MAPE	Parameter values
Single model	Ridge	0.064 2	0.052 1	0.101 1	ridge_alpha:1×10^{-3}
	Lasso	0.088 3	0.070 0	0.131 4	lasso_alpha:1×10^{-5}
	SVR	0.046 1	0.038 6	0.078 5	svr_c:1×10^{1}, svc_epsilon:1×10^{-2}
	Elastic_net	0.064 8	0.052 6	0.101 9	elastic_net_alpha:1×10^{-5}, elastic_net_l1_ratio: 5×10^{-2}
Stacking model	Stacking-ridge	0.021 0	0.015 3	0.032 8	lasso_alpha:1×10^{-4}, ridge_alpha:1×10^{-2}, elastic_net_alpha:1×10^{-4}, elastic_net_l1_ratio:5×10^{-2}, svr_c:1×10^{1}, svc_epsilon:1×10^{-5}
	Stacking-lasso	0.032 8	0.022 1	0.040 8	lasso_alpha:1×10^{-4}, ridge_alpha:1×10^{-2}, elastic_net_alpha:1×10^{-3}, elastic_net_l1_ratio:5×10^{-1}, svr_c:1×10^{1}, svc_epsilon:1×10^{-2}
	Stacking-SVR	0.025 4	0.022 7	0.053 7	lasso_alpha:1×10^{-4}, ridge_alpha:1×10^{-3}, elastic_net_alpha:1×10^{-4}, elastic_net_l1_ratio: 1×10^{-1}, svr_c:1×10^{1}, svc_epsilon:1×10^{-2}
	Stacking-elastic_net	**0.015 2**	**0.011 9**	**0.029 3**	lasso_alpha:1×10^{-4}, ridge_alpha:1×10^{-3}, elastic_net_alpha:1×10^{-4}, elastic_net_l1_ratio:5×10^{-2}, svr_c:1×10^{0}, svc_epsilon:1×10^{-3}

quantile of the best-fitting distribution. This method provides a probabilistic measure of confidence in the threshold setting, ensuring robust anomaly detection.

Assuming ϵ represents the component's tolerance, a component's parameters exceeding the ϵ range typically indicates a circuit anomaly. At this point, the FI value is defined as the anomaly threshold for that component. When parameters deviate by 3ϵ, it is considered that the circuit can no longer meet operational requirements, and the component is deemed to have reached the end of its life. Here, the FI value is set as the failure threshold for that component [23]. For the component R2↓, the calculated anomaly threshold is 0.4072, and the failure threshold is 0.5731. These thresholds are indicated in Fig. 5. Utilising historical FI data, the proposed algorithm predicts the future trend of FI and calculates the time required for FI to reach both the anomaly and failure thresholds. These times are used as the prediction intervals for fault warning and component failure, respectively.

Comparative experiments on model performance were conducted using lag features set at 10, with an equal split between the training and test sets, i.e. after training on the training set, recursive prediction was performed on the test set. The results of the fault prediction experiments for key components in the analog circuit are presented in Table 3, and the corresponding graphical representation is depicted in Fig. 6. It is evident that the predictive performance of the stacking models generally surpasses that of the

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

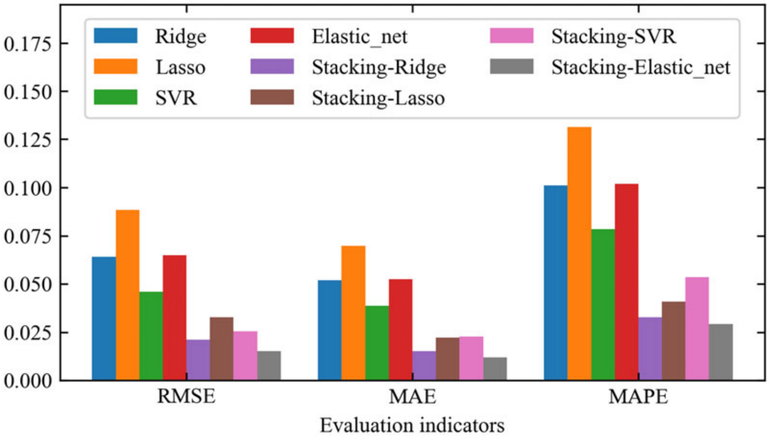

Figure 6. *Bar chart comparison of model performance.*

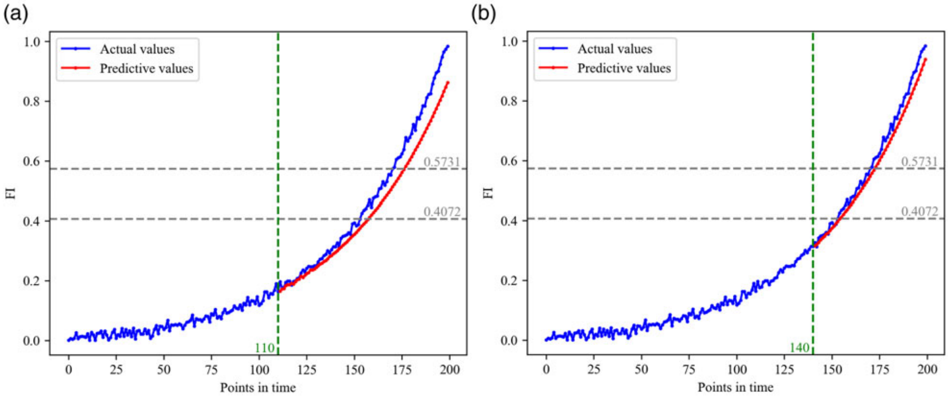

Figure 7. *Prediction curves with R2 decrease. (a) Using 110 training data point; (b) using 140 training data points.*

single models, with the stacking-elastic_net model achieving the best results across all three prediction metrics, thereby highlighting the superiority of this method.

For the top-performing stacking-elastic_net model, its predictive performance was evaluated using different training datasets. Specifically, for the component R2↓, the prediction curves for training datasets of 110 and 140 are illustrated in Fig. 7, to consider the feature changes over a longer time range, the lag features were set to 20 in both training data scenarios. It is observable that both training scenarios align well with the actual values. However, the prediction curve with a training dataset of 140 demonstrates a notably closer fit to the actual values, indicating enhanced accuracy.

The specific fault prediction errors are detailed in Table 4. The results indicate that with a training dataset of 110, the errors at the anomaly and failure points are 4 and 6, respectively. Conversely, with a training dataset of 140, the errors reduce to 1 and 2 at the anomaly and failure points, respectively, demonstrating a significantly improved prediction accuracy. This highlights the robust fault prediction capabilities of the stacking regression model.

4.2 Inertial measurement unit component fault prediction

The strapdown inertial navigation system (SINS) is an autonomous navigation system that operates independently of external information and is extensively utilised in aviation equipment. Its fundamental

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

Table 4. Prediction errors under R2↓

	Anomaly point			Failure point		
Training data count	Predicted anomaly point	Actual anomaly point	Error	Predicted failure point	Actual failure point	Error
110	158	154	4	177	171	6
140	155	154	1	173	171	2

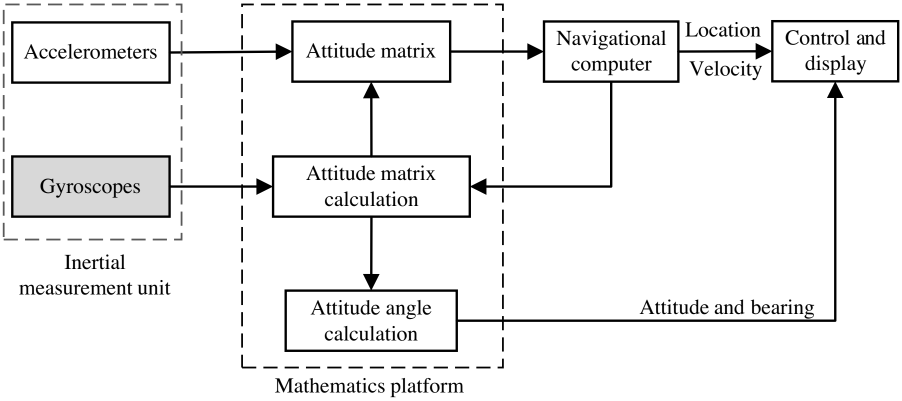

***Figure 8.** Basic principles of strapdown inertial navigation system.*

working principle is depicted in Fig. 8. The inertial measurement unit (IMU) is an essential electronic device within this system, tasked with measuring the angular velocity and linear acceleration of an aircraft. Typically, an IMU comprises three gyroscopes and three accelerometers, each dedicated to measuring the aircraft's angular velocity and linear acceleration along three respective axes. Through coordinate transformations and subsequent calculations, the position, velocity, and attitude angles of the aircraft are determined, enabling autonomous navigation. The performance of an IMU is primarily evaluated based on its measurement accuracy, which is significantly affected by the gyroscope drift coefficient. Consequently, the gyroscope drift coefficient is a vital performance metric for determining whether an IMU has deteriorated to the failure threshold.

The drift observed in our study is due to the gyroscope's performance degradation rather than poor updates of the Kalman filter. The IMU data used in our experiments were collected from controlled tests designed to simulate real-world fault scenarios, including seeded faults to ensure the reliability of the data. IMUs are known for their high reliability; however, for the purpose of this study, we introduced controlled faults to capture the degradation process accurately. This approach allows us to validate the fault prediction methods under realistic conditions.

This article utilises actual test data of the gyroscope from Ref. (24), selecting the first-order drift coefficient along the sensitive axis of the gyroscope as the primary index for monitoring performance. Through continuous operational tests, 96 data points of the first-order drift coefficient were collected at uniform sampling intervals. The experimental data, displayed in Fig. 9, distinctly show a nonlinear increasing trend.

Comparative experiments on model performance were conducted using a lag feature set at 10, with an equal ratio of training set to test set (1:1). After training on the training set, recursive predictions were performed on the test set. The results of the gyroscope prediction experiments are presented in Table 5, and the corresponding graphical representations are shown in Fig. 10. It is evident that the stacking regression model generally outperforms the single models, with the stacking-elastic_net model

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

Table 5. *Performance metrics comparison for gyroscope prediction models*

Model category	Model	RMSE	MAE	MAPE	Parameter values
Single models	Ridge	0.018 0	0.015 7	0.046 4	ridge_alpha:1×10^{-2}
	Lasso	0.016 2	0.014 0	0.041 2	lasso_alpha:1×10^{-4}
	SVR	0.023 4	0.020 6	0.060 3	svr_c:1×10^{0}, svc_epsilon:1×10^{-3}
	Elastic_net	0.010 7	0.009 1	0.027 0	elastic_net_alpha:1×10^{-3}, elastic_net_l1_ratio: 3×10^{-1}
Stacking models	Stacking-ridge	0.007 9	0.006 5	0.019 7	lasso_alpha:1×10^{-4}, ridge_alpha:1×10^{-2}, elastic_net_alpha:1×10^{-3}, elastic_net_l1_ratio:5×10^{-2}, svr_c:1×10^{1}, svc_epsilon:1×10^{-3}
	Stacking-lasso	0.014 5	0.012 1	0.034 7	lasso_alpha:1×10^{-3}, ridge_alpha:1×10^{-4}, elastic_net_alpha:1×10^{-3}, elastic_net_l1_ratio: 5×10^{-1}, svr_c:1×10^{-1}, svc_epsilon:1×10^{-2}
	Stacking-SVR	0.011 7	0.009 1	0.027 8	lasso_alpha:1×10^{-4}, ridge_alpha:1×10^{-3}, elastic_net_alpha:1×10^{-2}, elastic_net_l1_ratio:1×10^{-1}, svr_c:1×10^{1}, svc_epsilon:1×10^{-2}
	Stacking-elastic_net	**0.007 4**	**0.006 0**	**0.018 2**	lasso_alpha:1×10^{-4}, ridge_alpha:1×10^{-3}, elastic_net_alpha:1×10^{-3}, elastic_net_l1_ratio: 3×10^{-1}, svr_c:1×10^{0}, svc_epsilon:1×10^{-3}

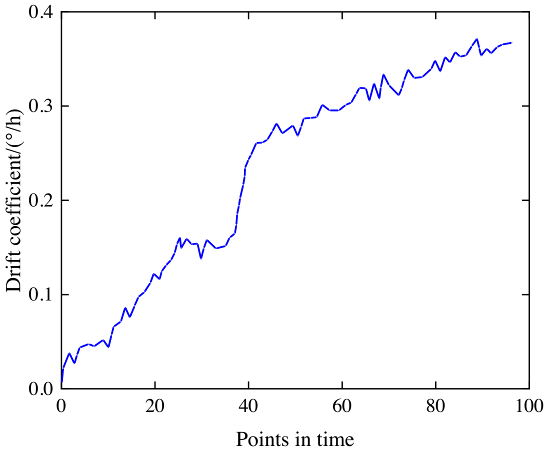

Figure 9. *Observed trend of gyroscope's first-order drift coefficient.*

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

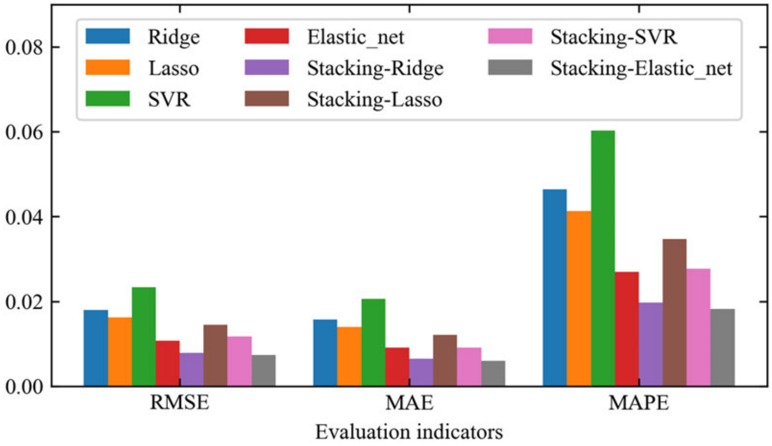

***Figure 10.** Graphical representation of model performance comparison.*

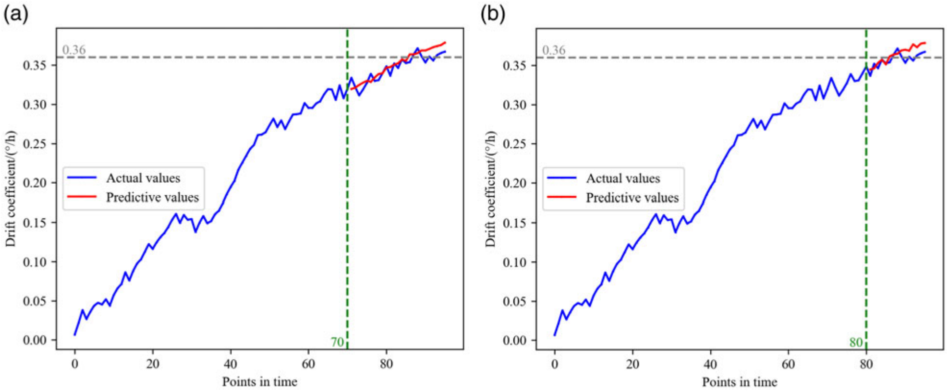

***Figure 11.** Gyroscope drift coefficient prediction curves. (a) Using 70 training data point; (b) using 80 training data points.*

achieving the best results across all three prediction metrics. This underscores the superior efficacy of the method.

For this type of gyroscope, a drift coefficient exceeding 0.36°/h is considered beyond the performance criteria, and thus, it is established as the fault threshold. The stacking-elastic_net model, noted for its superior prediction performance, was evaluated using different training datasets. The datasets were set to 70 and 80, respectively, with the remaining 26 and 16 data points serving as the test set to assess the algorithm's predictive capabilities. To consider the feature changes over a longer time range, the lag features were set to 20 in both training data scenarios. For gyroscope fault prediction, the prediction curves for the different training data volumes are illustrated in Fig. 11. It is observed that the prediction curves under both training data volumes show no significant differences, each aligning closely with the actual value curve.

The specific fault prediction errors are detailed in Table 6, showing that the prediction error time points for training data volumes of 70 and 80 are both 2, indicating relatively good predictive performance. From Fig. 11 and Table 6, it is clear that increasing the training data volume from 70 to 80 does not significantly improve the predictive performance. This lack of significant enhancement may

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

Table 6. *Prediction errors for gyroscope drift coefficient*

Training data count	Predicted anomaly point	Actual anomaly point	Error
70	86	88	2
80	86	88	2

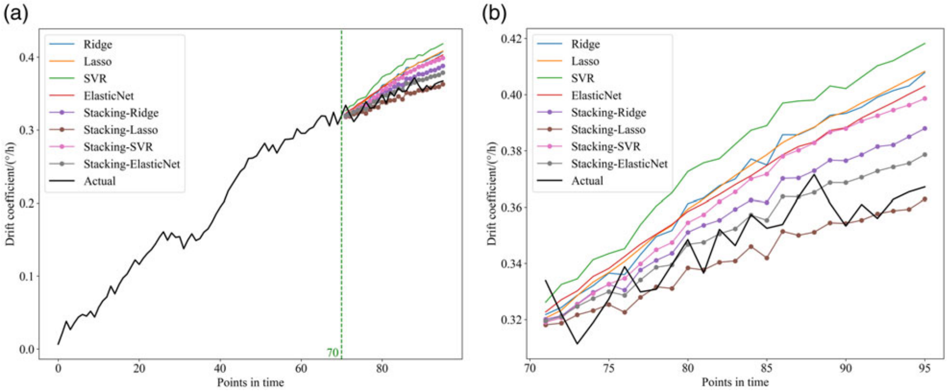

Figure 12. *Prediction curves for gyroscope drift coefficient by different methods. (a) Global prediction curve; (b) local prediction curve.*

be attributed to the strong irregularity present in the actual measured data of the gyroscope, which complicates the algorithm’s ability to accurately predict its trend.

With the training sample size set at 70 and the lag feature at 20, Fig. 12 displays the prediction curves and a detailed zoom-in on these curves for the gyroscope drift coefficient using all the methods mentioned previously. The figure clearly shows that the prediction curves of the stacking models are more closely aligned with the actual values compared to those of the single models. Among these, the stacking-elasticnet method exhibits the best fit to the degradation trend of the component performance. It is minimally affected by the accumulation of errors, thereby providing the most accurate fault prediction. This further underscores the superior performance of the stacking regression models.

5.0 Conclusion

This paper tackles the challenge of fault prediction for avionics components by proposing four multi-step prediction methods based on stacking regression. These methods integrate various standard regression models such as SVR, ridge regression, lasso regression, and elastic_net regression into a multi-level prediction framework. Within this framework, multiple base regression models in the first layer independently predict data, which are then fed into a meta-regressor in the second layer to leverage the predictive strengths of each base model. Additionally, the TPE algorithm was employed for hyperparameter optimisation, further enhancing the models’ performance. The results from fault prediction experiments on critical components of analog circuits and inertial measurement units demonstrate that the stacking regression models surpass traditional single models across several performance metrics. Notably, the stacking-elastic_net method exhibited the best predictive performance, affirming the effectiveness and practicality of the stacking regression approach in addressing complex fault prediction challenges in avionics components.

The fault prediction method for avionics components introduced in this paper not only enhances the accuracy of fault predictions but also offers an efficient technical solution for the health management of avionics. Future work will focus on exploring additional combinations of base and meta-regressors to further enhance predictive accuracy. We also aim to test the proposed methods in real-world avionics systems to validate their practical applicability. Additionally, integrating these models with Internet of Things (IoT) devices for real-time monitoring and prediction could open new avenues for research. Collaborations with industry partners will be sought to ensure the models are robust and scalable for various operational environments.

Data availability statement. The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments. We thank the authors of scikit-learn and Optuna open-source code. Our code is based on their open-source code for improvement.

Funding statement. This research was funded by the Mount Taishan Scholar Construction Project in Shandong Province, China, grant number: tstp20221146.

Competing interests. The authors declare no competing interest.

References

[1] Xue, Z., Yang, J., Chen, R., He, Q., Li, Q. and Mei, X. AR-assisted guidance for assembly and maintenance of avionics equipment, *Appl. Sci.*, 2024, **14,** (3), p 1137.

[2] Bertolino, A.C., De Martin, A., Jacazio, G. and Sorli, M. Design and preliminary performance assessment of a phm system for electromechanical flight control actuators, *Aerospace*, 2023, **10,** (4), p 335.

[3] Fu, S. and Avdelidis, N.P. Prognostic and health management of critical aircraft systems and components: An overview, *Sensors*, 2023, **23,** (19), p 8124.

[4] Hu, Y., Li, J., Hong, M., Ren, J. and Man, Y. Industrial artificial intelligence based energy management system: Integrated framework for electricity load forecasting and fault prediction, *Energy*, 2022, **244**, p 123195.

[5] Geng, S., and Wang, X. Predictive maintenance scheduling for multiple power equipment based on data-driven fault prediction, *Computers & Industrial Engineering*, 2022, **164**, p 107898.

[6] Chen, Z., O'Neill, Z., Wen, J., Pradhan, O., Yang, T., Lu, X. and Herr, T. A review of data-driven fault detection and diagnostics for building HVAC systems, *Appl. Energy*, 2023, **339**, p 121030.

[7] Liang, T.C. Fault prediction of avionics based on multi-depth belief network fusion, *Telecommun. Technol.*, 2021, **61,** (2), pp 248–253.

[8] Zhang, Q., Qi, J. and Jiao, H. Research on fault prediction method for instrument landing system based on GRU, *Adv. Aeronaut. Eng.*, 2024, **15,** (3), pp 62–70.

[9] Mitici, M., Hennink, B., Pavel, M. and Dong, J. Prognostics for Lithium-ion batteries for electric vertical take-off and landing aircraft using data-driven machine learning, *Energy AI*, 2023, **12**, p 100233.

[10] Gao, C., Li, B. and Dai, Z. Medium and long-term fault prediction of avionics based on echo state network, *Mobile Inf. Syst.*, 2022, **2022,** (1), p 5343909.

[11] Yoon, T. and Kang, D. Multi-modal stacking ensemble for the diagnosis of cardiovascular diseases, *J. Pers. Med.*, 2023, **13,** (2), p 373.

[12] Zhao, A.B. and Cheng, T. Stock return prediction: Stacking a variety of models, *J. Empirical Finance*, 2022, **67**, pp 288–317.

[13] Gu, J., Liu, S., Zhou, Z., Chalov, S.R. and Zhuang, Q. A stacking ensemble learning model for monthly rainfall prediction in the Taihu Basin, China, *Water*, 2022, **14,** (3), p 492.

[14] Lateko, A.A., Yang, H.T. and Huang, C.M. Short-term PV power forecasting using a regression-based ensemble method, *Energies*, 2022, **15,** (11), p 4171.

[15] Drucker, H., Burges, C.J., Kaufman, L., Smola, A., and Vapnik, V. Support vector regression machines, *Advances in Neural Information Processing Systems*, vol. 9, 1996.

[16] Hoerl, A.E. and Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics*, 1970, **12,** (1), pp 55–67.

[17] Tibshirani, R. Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 1996, **58,** (1), pp 267–288.

[18] Zou, H. and Hastie, T. Regularization and variable selection via the elastic net, *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 2005, **67,** (2), pp 301–320.

[19] Bergstra, J., Bardenet, R., Bengio, Y. and Kégl, B. Algorithms for hyper-parameter optimization, *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[20] Akiba, T., Sano, S., Yanase, T., Ohta, T. and Koyama, M. Optuna: A next-generation hyperparameter optimization framework, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp 2623–2631.

[21] Tang, X., Xu, A., Li, R., Zhu, M. and Dai, J. Simulation-based diagnostic model for automatic testability analysis of analog circuits, *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.*, 2017, **37,** (7), pp 1483–1493.

https://doi.org/10.1017/aer.2024.143 Published online by Cambridge University Press

[22] Li, M. Research on Key Technologies for Comprehensive Diagnosis and Fault Prediction of Complex Electronic Systems, Doctoral dissertation, University of Electronic Science and Technology of China, Chengdu, 2014.

[23] Zhu, M. Research on Key Technologies of PHM for Avionics Based on Extreme Learning Machine, Doctoral dissertation, Naval Aviation University, Yantai, 2019.

[24] Feng, L., Wang, H. and Zhou, Z. Online prediction of remaining useful life for inertial measurement units based on state space, *J. Tsinghua Univ. (Sci. Technol.)*, 2014, **54,** (4), pp 508–514.

Cite this article: Li W.H., Li G., Liu Y., Ma J.T., Wu Z.D., Tang X., Sun W.C. and Wen T.Z. Method of fault prediction for avionics components based on stacking regression. *The Aeronautical Journal*, https://doi.org/10.1017/aer.2024.143