

Article

An Efficient Task Implementation Modeling Framework with Multi-Stage Feature Selection and AutoML: A Case Study in Forest Fire Risk Prediction

Ye Su ^{1,2} , Longlong Zhao ^{1,*} , Hongzhong Li ¹ , Xiaoli Li ¹ , Jinsong Chen ^{1,3}  and Yuankai Ge ⁴

¹ Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; ye.su@siat.ac.cn (Y.S.); hz.li@siat.ac.cn (H.L.); xl.li2@siat.ac.cn (X.L.); js.chen@siat.ac.cn (J.C.)

² University of Chinese Academy of Sciences, Beijing 101407, China

³ Shenzhen Engineering Laboratory of Ocean Environmental Big Data Analysis and Application, Shenzhen 518055, China

⁴ The Eighth Engineering Co., Ltd., China Tiesiju Civil Engineering Group Co., Ltd., Hefei 230023, China; yk.ge@siat.ac.cn

* Correspondence: ll.zhao@siat.ac.cn; Tel.: +86-188-1008-0682

Abstract: As data science advances, automated machine learning (AutoML) gains attention for lowering barriers, saving time, and enhancing efficiency. However, with increasing data dimensionality, AutoML struggles with large-scale feature sets. Effective feature selection is crucial for efficient AutoML in multi-task applications. This study proposes an efficient modeling framework combining a multi-stage feature selection (MSFS) algorithm and AutoSklearn, a robust and efficient AutoML framework, to address high-dimensional data challenges. The MSFS algorithm includes three stages: mutual information gain (MIG), recursive feature elimination with cross-validation (RFECV), and a voting aggregation mechanism, ensuring comprehensive consideration of feature correlation, importance, and stability. Based on multi-source and time series remote sensing data, this study pioneers the application of AutoSklearn for forest fire risk prediction. Using this case study, we compare MSFS with five other feature selection (FS) algorithms, including three single FS algorithms and two hybrid FS algorithms. Results show that MSFS selects half of the original features (12/24), effectively handling collinearity (eliminating 11 out of 13 collinear feature groups) and increasing AutoSklearn's success rate by 15%, outperforming two FS algorithms with the same number of features by 7% and 5%. Among the six FS algorithms and non-FS, MSFS demonstrates the highest prediction performance and stability with minimal variance (0.09%) across five evaluation metrics. MSFS efficiently filters redundant features, enhancing AutoSklearn's operational efficiency and generalization ability in high-dimensional tasks. The MSFS–AutoSklearn framework significantly improves AutoML's production efficiency and prediction accuracy, facilitating the efficient implementation of various real-world tasks and the wider application of AutoML.



Citation: Su, Y.; Zhao, L.; Li, H.; Li, X.; Chen, J.; Ge, Y. An Efficient Task Implementation Modeling Framework with Multi-Stage Feature Selection and AutoML: A Case Study in Forest Fire Risk Prediction. *Remote Sens.* **2024**, *16*, 3190. <https://doi.org/10.3390/rs16173190>

Academic Editor: Hossein M. Rizeei

Received: 5 June 2024

Revised: 22 August 2024

Accepted: 26 August 2024

Published: 29 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Amid the booming development of data science, the successful application of machine learning (ML) in various fields has garnered extensive attention and recognition. However, the traditional ML workflow often involves numerous complex and time-consuming steps, such as data preprocessing, feature engineering, model selection, and hyperparameter tuning. These processes require operators to possess profound ML expertise and extensive practical experience. For non-ML expert users, these complex tasks can be daunting, and manual model tuning can be a tedious and time-consuming challenge even for experienced ML professionals [1,2].

Automated machine learning (AutoML) is a technique that automates the end-to-end process of applying ML to real-world tasks. Its goal is to achieve the automatic execution of a series of tasks such as data preprocessing, model selection, and tuning, significantly simplifying the ML workflow, lowering the application barriers, and enhancing the ability to deploy model algorithms in practical applications [3–5]. Furthermore, AutoML leverages optimization algorithms such as Bayesian optimization and random search to intelligently search for the best model configurations and hyperparameters, further improving model performance while achieving automatic learning [6–8]. However, the accelerated growth of data science has led to an increase in the volume of high-dimensional data, limiting the further application of AutoML in practical needs [9]. Irrelevant and redundant features present in high-dimensional data increase the time and space complexity of models, leading AutoML to require substantial computational resources and longer runtimes. This increases the time and computational costs, thereby seriously affecting the accuracy and efficiency of the models. Facing the challenges posed by high-dimensional data, it is crucial to select effective features with a high contribution and stability to ensure the efficient operation of AutoML in multi-task applications.

Feature selection (FS) is an effective method to simplify ML models, capable of reducing feature dimensionality [10,11]. FS selects the most representative features from the original feature set that can provide sufficient information by eliminating redundant and irrelevant features. This process compresses the search space of learning algorithms, thereby improving model learning speed and performance [12,13]. Combined with FS algorithms, AutoML can better tackle the challenges posed by large-scale datasets. It reduces computational resource consumption, accelerates the model training process, improves the success rate of model training, enhances the diversity of model selection, and then facilitates the efficient implementation of real-world tasks.

FS algorithms can be classified into three categories: filter, wrapper, and embedded [14–17]. Filter methods select features based on statistical or information-theoretic metrics [18,19], such as mutual information (MI), which can pick out features strongly correlated with the target variable but tend to overlook the correlation between features [20]. Wrapper methods integrate feature selection into model training, such as sequential forward selection (SFS) [21,22], which can consider complex relationships between features but are prone to overfitting and ignore the correlation between individual features and the target variable [23,24]. Embedded methods automatically select features during the model training process, such as Lasso, which has good flexibility and generalization ability but whose performance is significantly influenced by the model [25]. However, these single FS algorithms often struggle to simultaneously meet multiple requirements, such as prediction accuracy, computational efficiency, and stability.

To address the aforementioned issues, hybrid feature selection (HFS) algorithms have been proposed, which combine the advantages of multiple single FS algorithms to achieve a higher prediction accuracy and lower computational cost [26–30]. For instance, the mutual information decision tree regressor (MIDTR) integrates filter-based and embedded-based algorithms, considering both the correlation between features and their correlation to the target variable [31]. The improved recursive feature elimination with cross-validation (ImprovedRFECV) combines wrapper-based and embedded-based algorithms, considering both the complex relationships between features and the stability and scalability of the algorithm [32]. While these HFS algorithms can alleviate some of the issues of single algorithms to a certain extent, their application within AutoML frameworks still faces numerous challenges. For example, the MIDTR algorithm fails to effectively address collinearity issues among features, and its stability has not been verified. The ImprovedRFECV algorithm neglects the correlation between individual features and the target variable. Moreover, AutoML requires the screening and tuning of a large number of models to enhance the diversity of model selection and increase the probability of successful model operation. This imposes higher demands on the effectiveness and stability of feature subsets. Therefore, it is crucial to construct a FS algorithm that can comprehensively consider the correlation

between features and the target variable, as well as the interrelationships between features, while maintaining a high stability. This will enable AutoML to efficiently process high-dimensional datasets, thus facilitating the efficient implementation of complex tasks.

Forest fires are characterized by their suddenness, rapid spread, and catastrophic nature, posing high demands on the prediction accuracy and timeliness of risk prediction models [33,34]. Meanwhile, the occurrence of forest fires is influenced by a complex interplay of environmental factors, such as meteorology, terrain, vegetation, and human activities [35–39]. With the continuous emergence of multi-source heterogeneous remote sensing, meteorological, basic geographic information, and socio-economic data, the number of environmental factors used for forest fire risk prediction is steadily increasing. Selecting a subset of features with a high contribution and stability from this high-dimensional space is a significant challenge in achieving efficient modeling for fire prediction. Furthermore, forestry department managers who implement fire risk warnings and formulate prevention measures often do not possess professional ML knowledge. Therefore, forest fire risk prediction becomes an ideal application domain for combining FS algorithms and AutoML.

In summary, to address the urgent need for feature selection in AutoML to ensure its efficient operation, this study uses a forest fire as a case study and integrates AutoSklearn, a robust and efficient AutoML framework [9], to conduct research. This study marks the first application of AutoML in the field of fire risk warning, aiming to improve the accuracy and efficiency of forest fire risk prediction models while promoting the efficient application of algorithmic technology to practical needs, thus providing feasible technical support for forest fire prevention. The specific objectives include the following: (1) proposing a FS algorithm that comprehensively considers the correlation between features and target variables, inter-feature correlations, and stability, to yield an optimal feature subset supporting the efficient operation of AutoML; (2) conducting comparative experiments by combining various FS algorithms with AutoSklearn to validate the effectiveness and stability of the proposed FS algorithm; and (3) thoroughly analyzing the advantages of AutoSklearn combined with the proposed FS algorithm and clarifying its application prospects for efficiently implementing various application tasks.

2. Study Area and Data Sources

2.1. Study Area

The study area comprises Guangdong, Hong Kong, and Macao in southeastern China (Figure 1), characterized by a subtropical monsoon climate with distinct wet and dry seasons. The rainy period spans from April to October, with intense rainfall in summer and dryness in winter. Guangdong boasts a diverse topography with extensive mountainous and hilly areas. As of 2022, statistics indicate that Guangdong has a forest area of 1.05 million hectares, with a forest coverage rate reaching 58.7%, making it one of the greenest provinces in China. In recent years, with the deepening of Guangdong's new round of greening initiatives, including the closure of mountains for afforestation, returning farmland to forests, artificial afforestation, and strict forest fire prevention measures, the accumulation of combustible materials has continuously increased. In some forest areas, the critical threshold of combustible material load, which can spark forest fires, has been reached or surpassed at 30 tons per hectare [40], posing a considerable challenge to forest fire prevention. Furthermore, extreme weather events stemming from climate change, such as heatwaves and droughts, have further aggravated the risk of forest fires in the study area. According to statistical data, from 2018 to 2022, Guangdong experienced an average of approximately 130 forest fires annually, affecting an average of about 800 hectares of forest area each year. The economic losses resulting from these fires exceeded 50 million yuan annually [41], posing a severe threat to local economic stability and ecological security.

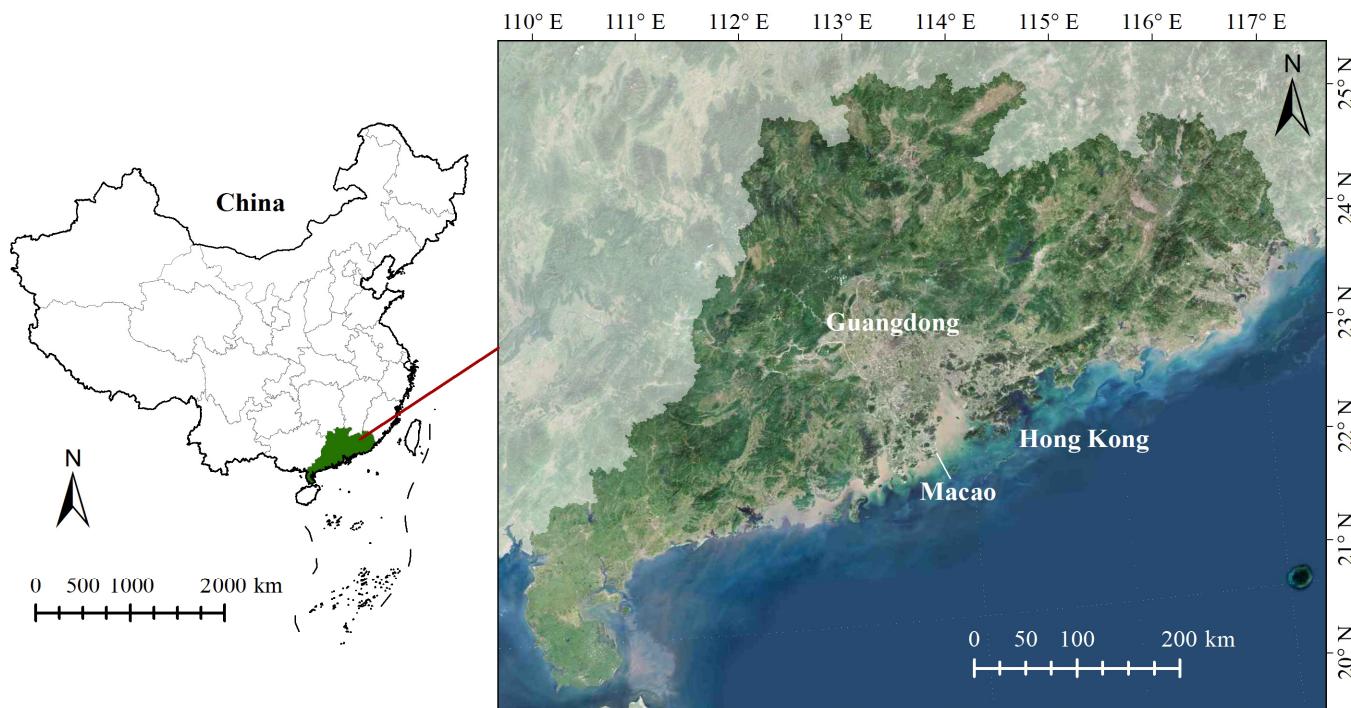


Figure 1. Study area.

2.2. Data Source and Processing

In this study, the data sources used mainly include three categories: fire data, remote sensing products, and remote sensing images. The detailed information on each data source is shown in Table 1.

The FIRMS VIIRS fire data include attributes of latitude and longitude, brightness temperature, date of occurrence, confidence level, fire type, etc. Using the forest distribution data provided by the CLCD product, the fire point data were cleaned and filtered to identify 2175 forest fire points (FPs) within the study area from 2015 to 2019. Additionally, based on the space-humidity-constrained negative samples sampling method proposed in [42], 10,003 pseudo fire points (PFPs) were generated for the period from 2015 to 2019. Thus, a sample set of forest fire risk predictions in the study area was constructed for model training and validation.

The study used four types of remote sensing products, namely a digital elevation model (DEM), the China land cover dataset (CLCD), the climate hazards group infrared precipitation with stations dataset (CHIRPS), and a land surface temperature product (MYD11A1). The DEM was used to generate terrain factor features such as elevation, slope, and aspect in the study area. The CLCD was used to provide information on forest distribution ranges, assisting in sample set construction. The CHIRPS dataset provided precipitation data was used to generate PFPs and precipitation features of meteorological factors. The LST product was employed to extract the land surface temperature feature of meteorological factors, converting the units from K to °C.

The remote sensing images mainly consisted of MODIS land surface reflectance data (MYD09GA), which were used to calculate vegetation indices such as the normalized difference vegetation index (NDVI), the enhanced vegetation index (EVI), and the normalized difference water index (NDWI) for forests. The NDVI and EVI were used to indicate changes in vegetation greenness, while the NDWI was used to indicate vegetation water content. Although the NDVI is widely recognized as one of the most effective parameters for representing vegetation, it has a low sensitivity to high-density vegetation and is prone to saturation. The EVI can mitigate this issue [43]. Therefore, this study employed both the NDVI and EVI to indicate vegetation greenness.

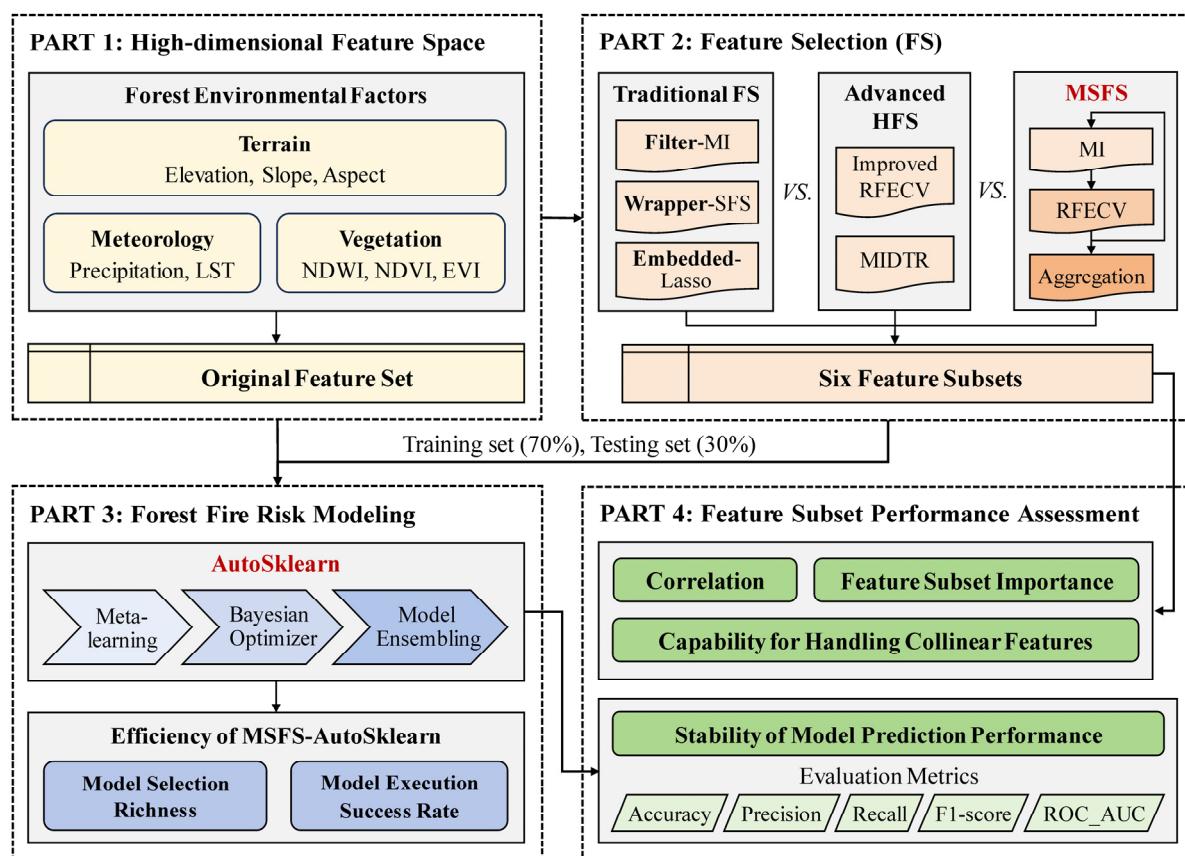
Table 1. Data sources and their details and usage.

Data Type	Data Name	Sources	Spatial Resolution	Temporal Resolution	Time Range	Usage
Fire data	VIIRS	NASA FIRMS	375 m	Daily	2015–2019	Sample set construction
	DEM	NASA SRTM	30 m	-	2000	Terrain feature extraction
Remote sensing product	CLCD	[44]	30 m	Yearly	2015–2019	Sample data cleaning
	CHIRPS	UCSB/CHC	5566 m	Daily	2015–2019	Sample set and feature set construction
Satellite image	MYD11A1	USGS	1000 m	Daily	2015–2019	LST feature extraction
	MYD09GA	USGS	500 m	Daily	2015–2019	Vegetation feature extraction

Note: VIIRS: visible infrared imaging radiometer suite; DEM, digital elevation model; CLCD, China land cover dataset; CHIRPS, climate hazards group infrared precipitation with stations; NASA, National Aeronautics and Space Administration; FIRMS, fire information for resource management system; SRTM, shuttle radar topography mission; UCSB/CHG, Climate Hazards Center at the University of California; USGS, United States geological survey.

3. Methods

The methodology of this study comprises three main parts: (1) constructing a high-dimensional feature space for forest fire risk prediction, (2) developing a multi-stage feature selection (MSFS) algorithm and selecting comparison algorithms, and (3) modeling forest fire risk prediction using AutoSklearn. The methodology flowchart is depicted in Figure 2.

**Figure 2.** Methodology flowchart.

3.1. High-Dimensional Feature Space Construction

Considering the accessibility of data, this study selected three types of factors indicative of a forest fire-prone environment to construct the feature space for forest fire prediction: terrain (elevation, slope, and aspect), meteorological conditions (precipitation and surface temperature), and vegetation (NDVI, EVI, and NDWI) (Table 2). Specifically, in extracting the five features of dynamic environmental factors of meteorology and vegetation that change over time, the method based on dynamic time windows (DTWs) proposed in [42] was adopted to obtain more refined information on the spatial heterogeneity of the cumulative dryness state of the forest. This method determines the DTWs for acquiring the features of dynamic environmental factors by setting a cumulative precipitation threshold ($Thcp$) and a daily precipitation threshold ($Thdp$). Subsequently, four statistical synthesis values (min, median, mean, and max) were calculated for the five dynamic environmental factors within the DTWs, with an additional summation value for precipitation. These statistical methods were adopted to include features that could best reflect the dryness state of forest fuels. We set $Thcp$ to 30 mm and $Thdp$ to 10 mm, thus extracting 21 features for the five dynamic environmental indicators mentioned above. Ultimately, the high-dimensional feature space constructed in this study contains 24 features of both dynamic and static environmental factors, as detailed in Table 2.

Table 2. High-dimensional feature space involving static and dynamic environmental factors.

Environmental Factors	Indicators	Features
Terrain (static)	1. Elevation 2. Slope 3. Aspect	(1) Elevation (2) Slope (3) Aspect
Vegetation (dynamic)	4. NDWI 5. NDVI 6. EVI	(4) NDWI _{max} ; (5) NDWI _{mean} ; (6) NDWI _{median} ; (7) NDWI _{min} (8) NDVI _{max} ; (9) NDVI _{mean} ; (10) NDVI _{median} ; (11) NDVI _{min} (12) EVI _{max} ; (13) EVI _{mean} ; (14) EVI _{median} ; (15) EVI _{min}
Meteorology (dynamic)	7. Precipitation 8. LST	(16) PREC _{max} ; (17) PREC _{mean} ; (18) PREC _{median} ; (19) PREC _{min} ; (20) PREC _{sum} (21) LST _{max} ; (22) LST _{mean} ; (23) LST _{median} ; (24) LST _{min}

3.2. Feature Selection

To effectively avoid feature dimensionality expansion, reduce redundant features, decrease the computational burden, and enhance the generalization capability of the model, this study proposes a multi-stage feature selection (MSFS) algorithm that can obtain the optimal feature subset in stages based on a series of FS algorithms. To validate the performance of the MSFS algorithm, five FS algorithms were selected for comparative experiments, including three traditional FS algorithms: the filter algorithm MI, the wrapper algorithm SFS, and the embedded algorithm Lasso, as well as two advanced HFS algorithms: ImprovedRFECV and MIDTR. The principles of these five algorithms and the MSFS algorithm are described below.

3.2.1. Traditional Feature Selection Algorithms

The MI algorithm is a metric that measures the amount of shared information between two variables [45,46]. In the context of forest fire risk prediction, the relationship between features and fire risk is crucial. MI can accurately quantify the degree of dependence between each feature and fire risk, thereby assisting in identifying the features that have the most significant impact on the prediction results.

SFS is an algorithm designed to iteratively construct an optimal feature subset. It does so by sequentially adding the feature that contributes most to enhancing the model's predictive performance until a predefined criterion is satisfied, thereby facilitating efficient and precise feature selection [47]. The environmental factors related to forest fires exhibit complex relationships and interactions. In each iteration, SFS assesses the combined impact

of newly introduced features with the existing ones, ensuring that the final selected feature subset comprehensively captures these complex relationships and consequently enhances the generalization capabilities of the prediction model.

The Lasso algorithm uses an L1 regularization term to impose constraints on model coefficients, which potentially results in the reduction in some coefficients to zero [48]. The various features associated with the environmental factors of forest fires often exhibit high correlations, posing challenges to model stability and prediction accuracy. By compressing coefficients and eliminating redundant features, Lasso can effectively address this issue of multicollinearity, thereby enhancing the stability and prediction accuracy of the model.

3.2.2. Advance Hybrid Feature Selection Algorithms

The ImprovedRFECV algorithm is an enhanced version of the traditional recursive feature elimination with the cross-validation (RFECV) algorithm [32]. It employs multiple strategies to address issues inherent in the traditional RFECV algorithm, such as susceptibility to accuracy and stability disruptions due to noisy data, and the over-reliance on the chosen model, which can affect performance and stability [32,49]. These strategies include using repeated random sampling methods to increase the stability of the optimal feature subset, introducing L1 and L2 regularization to mitigate the impact of noise and collinearity issues, and adopting a multi-model ensemble learning framework to leverage the strengths of multiple models and avoid dependence on any single model, thereby enhancing the algorithm's generalization capability. As a result, in the context of forest fire risk prediction, the ImprovedRFECV algorithm can comprehensively consider various factors that may influence fire risk, thereby improving the accuracy and stability of the predictions.

The MIDTR algorithm combines mutual information and decision tree algorithms to identify an optimal feature subset. Initially, it filters out 25% of the features based on mutual information. Subsequently, an additional 25% of the features are eliminated through a decision tree algorithm, ultimately leading to the selection of an optimal feature subset [31]. This algorithm leverages the noise tolerance of decision trees, preventing information loss that may arise from overly stringent mutual information filtering. Furthermore, it considers both the correlations between features and their correlations to the target variable, enabling a more comprehensive identification of the key factors that influence forest fire risk, thereby enhancing prediction accuracy.

3.2.3. Multi-Stage Feature Selection Algorithm: MSFS

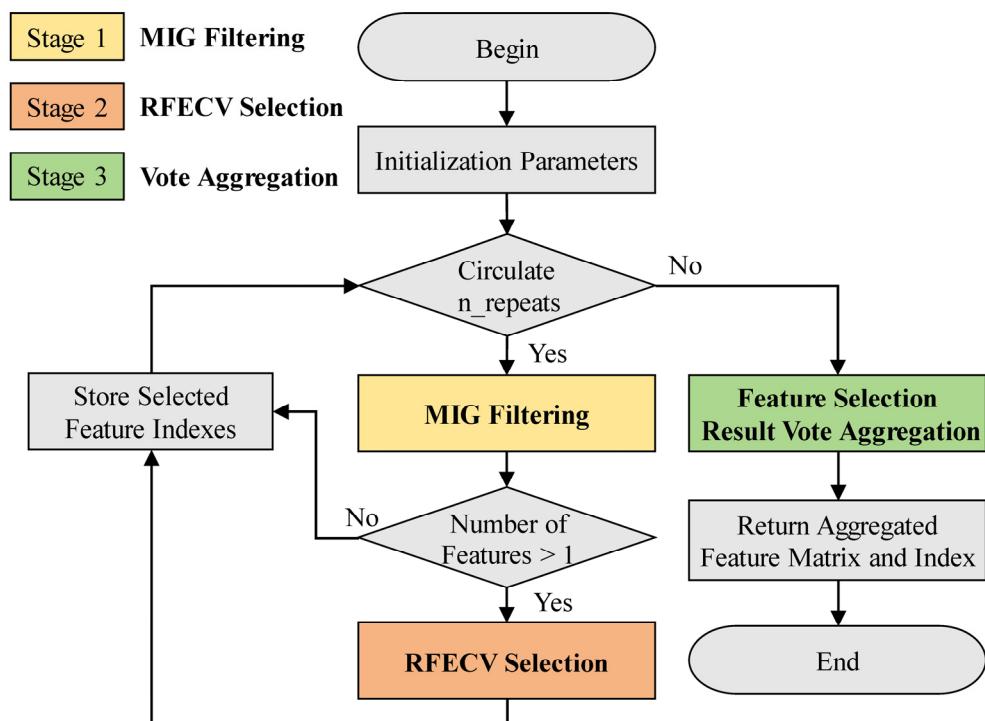
To identify the most contributory and stable feature subset, we propose the MSFS algorithm, comprising three distinct stages. The first stage, MIG filtering, considers the correlation between individual features and the target variable. The second stage, RFECV selection, incorporates the inter-feature correlations into the selection process. The third stage, a voting aggregation mechanism, ensures that the selected feature subset optimizes model stability. The pseudocode of the MSFS algorithm is presented in Algorithm 1.

The MSFS algorithm iteratively performs the feature selection process. In each iteration, it performs two steps: MIG filtering (stage 1) and RFECV selection (stage 2), resulting in a set of selected features. However, RFECV filtering is only executed if the number of features chosen in stage 1 exceeds one; otherwise, the algorithm proceeds to the next iteration by saving the indices of the stage 1 feature subset. This cycle continues until $n_{repeats}$ iterations are completed, with the indices of the feature subsets from each iteration being compiled in a list. Ultimately, the optimal feature subset is identified through a voting aggregation mechanism (stage 3). Figure 3 depicts the flowchart of the MSFS algorithm, while the subsequent sections elaborate on the detailed processes of its three stages.

Algorithm 1 MSFS Algorithm

Input: feature matrix, target variable, number of features to select using mutual information, estimator for RFECV, number of cross-validation folds, number of repetitions for increased stability
 Output: final selected feature set, indices of the final selected features

- 1: Initialize selected_feature_indices_all_runs as an empty list
- 2: For each repetition from 1 to n_repeats:
- 3: $X_{mi}, \text{selected_feature_indices}_{mi} \leftarrow \text{MIG}(X, y, k_{mi})$
- 4: If the number of features in X_{mi} is greater than 1:
- 5: $X_{rfecv}, \text{selected_feature_indices}_{rfecv} \leftarrow \text{RFECV}(X_{mi}, y, \text{estimator}, cv)$
- 6: selected_feature_indices $\leftarrow \text{selected_feature_indices}_{mi}[\text{selected_feature_indices}_{rfecv}]$
- 7: Else:
- 8: selected_feature_indices $\leftarrow \text{selected_feature_indices}_{mi}$
- 9: $X_{rfecv} \leftarrow X_{mi}$
- 10: Add selected_feature_indices to the selected_feature_indices_all_runs list
- 11: Initialize feature_votes as a zero array of length equal to the number of features in X
- 12: For each feature_indices in selected_feature_indices_all_runs:
- 13: Increment the votes in feature_votes at the indices corresponding to feature_indices
- 14: Initialize selected_feature_indices_final as an empty list
- 15: For each index i in feature_votes:
- 16: If feature_votes[i] is greater than half of n_repeats:
- 17: Add i to the selected_feature_indices_final list
- 18: $X_{final} \leftarrow \text{The columns of } X \text{ corresponding to the indices in selected_feature_indices}_{final}$
- 19: Return X_{final} and selected_feature_indices_final

**Figure 3.** Flowchart of MSFS algorithm.

1. MIG Filtering

- ① For $\forall f \in F$, calculating the MIG between f and target variable t , and selecting a subset of features F' strongly related to t is performed, as shown in Equation (1).

$$\text{MIG}(f) = I(f; t) - I(t) \quad (1)$$

where F represents the set of features; f is a single feature in F ; $MIG(f)$ denotes the MIG of feature f ; $I(f; t)$ stands for the mutual information between f and t ; and $I(t)$ represents the self-information of t when only t is considered, i.e., the entropy of t . In feature selection, we generally disregard $I(t)$ as it is a constant and does not depend on the feature f . Therefore, the MIG can also be simplified to Equation (2).

$$MIG(f) = I(f; t) \quad (2)$$

The calculation formula for $I(f; t)$ is shown in Equation (3).

$$I(f; t) = \sum_{f \in F} \sum_t P(f, t) \log \frac{P(f, t)}{P(f)P(t)} \quad (3)$$

where $P(f, t)$ represents the joint probability distribution of f and t ; $P(f)$ and $P(t)$ are the marginal probability distributions of f and t , respectively.

- ② Return the feature matrix filtered by MIG and the indices of the selected features.
- 2. RFECV Selection
 - ① The initial feature set F' is obtained through MIG filtering.
 - ② Modeling is performed using F' , and then the importance of each feature is calculated.
 - ③ Recursively remove the least important feature and update the feature set.
 - ④ Go back to step ② until the importance of all features has been rated.
 - ⑤ Based on the feature importance determined during the RFE phase, select different numbers of features sequentially.
 - ⑥ Perform cross-validation on the selected feature sets.
 - ⑦ Determine the number of features with the highest average score to complete the feature selection.
 - ⑧ Return the feature matrix selected by RFECV and the indices of the selected features.
- 3. Vote Aggregation Mechanism

During the aggregation process, the number of times each feature is selected (i.e., the vote count) is calculated. Subsequently, only those features that are selected in most runs (with a vote count exceeding $n_repeats/2$) are retained as the final selected features. The following outlines the specific steps of this process.

- ① Initialize an array A of zeros with the same length as the original feature count.
- ② Iterate through the indices of the selected features in each run of L and cast votes (i.e., increment the count) for the corresponding positions in A .
- ③ Select features based on the voting results: choose features that are selected in more than half of the runs (i.e., A values greater than $n_repeats/2$).
- ④ Return the aggregated feature matrix and the indices of the selected features.

3.3. Forest Fire Risk Modeling

To validate the effectiveness of the proposed MSFS algorithm in promoting the efficient operation of AutoML, this study modeled forest fire risk based on the AutoSklearn algorithm, an efficient and robust AutoML framework. The principles of the AutoSklearn algorithm, along with the modeling process and parameter configuration, are outlined below.

3.3.1. AutoSklearn Algorithm

The AutoSklearn algorithm efficiently automates the selection of appropriate algorithms, feature preprocessing steps, and their respective hyperparameters for new datasets. The core of AutoSklearn lies in automating the machine learning workflow, which is accomplished through a combination of three techniques: meta-learning, Bayesian optimizer, and

model ensembling. This algorithm can automatically discover the optimal model pipeline and corresponding hyperparameters [9]. The principle of the AutoSklearn algorithm is illustrated in Figure 4.

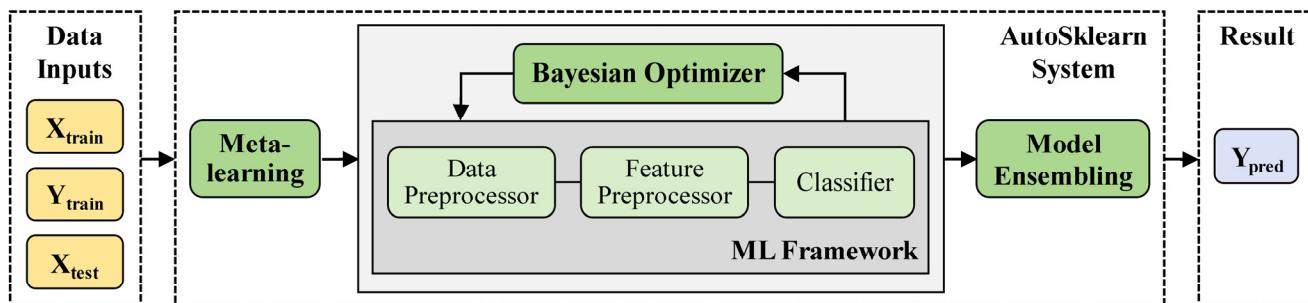


Figure 4. Principal diagram of AutoSklearn system (modified from [9]). X_{train} , training feature set; Y_{train} , training label set; X_{test} , test feature set; and Y_{pred} , AutoSklearn prediction result.

1. Meta-learning

AutoSklearn initially uses meta-learning to discover optimal hyperparameter configurations for Bayesian optimization, thereby surpassing random performance from inception. Meta-learning aims to discern patterns where datasets with comparable meta-features exhibit similar performance under the same hyperparameters [50]. These meta-features represent effective computational attributes of the dataset, assisting in determining the appropriate algorithm for a new dataset. During offline training, AutoSklearn records the hyperparameters that yielded the best results for each reference dataset. These hyperparameters are then used as initialization for the Bayesian optimizer when processing new datasets with similar meta-features.

2. Bayesian Optimizer

AutoSklearn employs Bayesian optimization to identify the optimal model pipeline. Within a defined search space, it explores the best combination of model pipelines and hyperparameters. This search space comprises diverse machine learning algorithms, including decision trees, random forest, support vector machines, neural networks, etc., along with their respective hyperparameters, such as learning rate and tree depth, etc. The Bayesian optimizer gradually approximates the optimal solution by evaluating various combinations and monitoring their performance on a validation set [51].

The principle of Bayesian optimization lies in leveraging the performance of existing samples within the objective function to construct a posterior model [52]. Each point on this posterior model represents a Gaussian distribution, with a mean and variance. Specifically, for known sample points, the mean corresponds to the objective function's value at that point, with a variance of 0. Conversely, for unknown sample points, AutoSklearn uses an acquisition function to iteratively probe the objective function's values, thereby updating the posterior probability model. During the Bayesian optimization procedure, AutoSklearn incorporates meta-learning for warm-starting optimization [53–55]. This approach entails that if a particular model or hyperparameter combination exhibits promising performance in prior searches, AutoSklearn will more frequently attempt this combination in new searches. This strategy not only accelerates the search process but also helps to avoid local optima.

3. Automated ensemble of models

Upon completion of the search process, AutoSklearn produces one or more optimized model pipelines that incorporate diverse machine learning algorithms and hyperparameter configurations. To bolster predictive performance, AutoSklearn automatically ensembles these models. Specifically, it assigns weights to the predictions of multiple models, optimizing these weights through techniques including stacking [56], gradient-free numerical

optimization, and the method ensemble selection [57], thereby yielding a more robust and accurate prediction result.

3.3.2. AutoSklearn Modeling and Parameter Configuration

During the modeling of forest fire risk, PFPs were sampled at a 1:1 ratio with FPs. A total of 70% of the dataset was used for training, and the remaining 30% was used for testing. AutoSklearn was configured with five key parameters (Table 3). Among these, the ‘time_left_for_this_task’ parameter determined the total time (in seconds) allotted for the entire automated machine learning process, set to 600 s (10 min) in this study. During this time limit, AutoSklearn explored various model configurations and hyperparameters to identify the optimal model. Additionally, the ‘per_run_time_limit’ parameter imposed a training time limit (in seconds) for each model configuration or hyperparameter combination, set to 30 s for this study. If a configuration’s training time exceeded this limit, AutoSklearn terminated the training and moved on to evaluate other configurations. This parameter efficiently explored numerous model configurations within the specified time constraint. The ‘metric’ parameter dictated the performance criterion for model selection and evaluation. We chose accuracy as the evaluation metric. The ‘seed’ parameter was used to establish a reproducible random seed, ensuring consistency in model selection and hyperparameter configuration across multiple runs of AutoSklearn. The ‘resampling_strategy’ parameter denoted the approach employed for model selection and evaluation. The ‘cv’ (cross-validation) method was adopted to assess model performance. Cross-validation involves partitioning the dataset into multiple subsets, sequentially using them as training and test sets, thereby evaluating the model’s generalization capabilities across different data segments. The ‘resampling_strategy_arguments’ parameter further specified additional parameters related to the resampling strategy. In this instance, the number of folds (or subsets) for cross-validation was set to 5, indicating that the dataset was divided into five subsets, which AutoSklearn then used for a 5-fold cross-validation procedure.

Table 3. Parameter configuration table for AutoSklearn.

Parameters	Value
time_left_for_this_task	600
per_run_time_limit	30
metric	autosklearn.metrics.accuracy
seed	42
resampling_strategy	‘cv’
resampling_strategy_arguments	{‘folds’: 5}

3.4. Model Performance Assessment

3.4.1. Evaluation Metrics

To comprehensively evaluate the performance of the forest fire risk prediction model, five evaluation metrics were used: accuracy, precision, recall, the F1-score (Equations (4)–(7)), and the ROC_AUC. Accuracy quantifies the overall accuracy of the model’s predictions. Precision assesses the model’s ability to correctly identify positive samples (fire points) by measuring the proportion of true positives among predicted positives. Recall focuses on the proportion of true positive samples that the model successfully identifies. The F1-score, being the harmonic mean of precision and recall, provides a balanced reflection of both metrics’ performance. Furthermore, the ROC curve was introduced to evaluate the classification effectiveness of the model by plotting the false positive rate (FPR) against the true positive rate (TPR) across various probability thresholds. The area under this curve, ROC_AUC, serves as a stable quantitative measure of model performance. Values for these five metrics range from 0 to 1, with higher values indicating superior model prediction performance. To ensure fairness and accuracy in evaluation, we executed ten experiments, resampling ten times for samples. Each resampling used 5-fold cross-validation for training,

and the arithmetic mean of the outcomes was adopted as the definitive score for each evaluation metric.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (4)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

$$\text{F1-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

where TP (true positive) denotes the count of FPs correctly predicted as FPs, FP (false positive) represents the count of PFPs falsely predicted as FPs, FN (false negative) signifies the count of FPs falsely predicted as PFPs, and TN (true negative) indicates the count of PFPs correctly predicted as PFPs.

3.4.2. Environment Setting

This study was conducted on the Kaggle platform (<https://www.kaggle.com/> (accessed on 27 February 2024)), a globally renowned platform for data science and machine learning competitions. The experimental environment was configured with a Linux system, running version 5.15.133+. The Python interpreter was version 3.7.12, compiled by GCC 9.4.0. Additionally, various Python packages were installed, including ConfigSpace-0.4.21, AutoSklearn-0.15.0, distro-1.9.0, emcee-3.1.4, liac-arff-2.5.0, pynisher-0.6.4, pyrfr-0.8.3, scikit-learn-0.24.2, and smac-1.2.

4. Results

4.1. Feature Subsets Based on Six FS Algorithms

4.1.1. Number and Contributions of Six Feature Subsets

To assess the dimensionality reduction capabilities of six FS algorithms and the correlation of their resulting feature subsets with fire risk, we visualized the statistical data obtained from six FS algorithms, as illustrated in Figure 5. From Figure 5a, the Lasso algorithm exhibited the most prominent dimensionality reduction, selecting only four features. The MIDTR, ImprovedRFECCV, and MSFS algorithms each retained half of the original features (12), demonstrating their high dimensionality reduction capabilities. Conversely, the SFS and MI algorithms selected 15 and 16 features, respectively, suggesting a relatively poor dimensionality reduction compared to other algorithms.

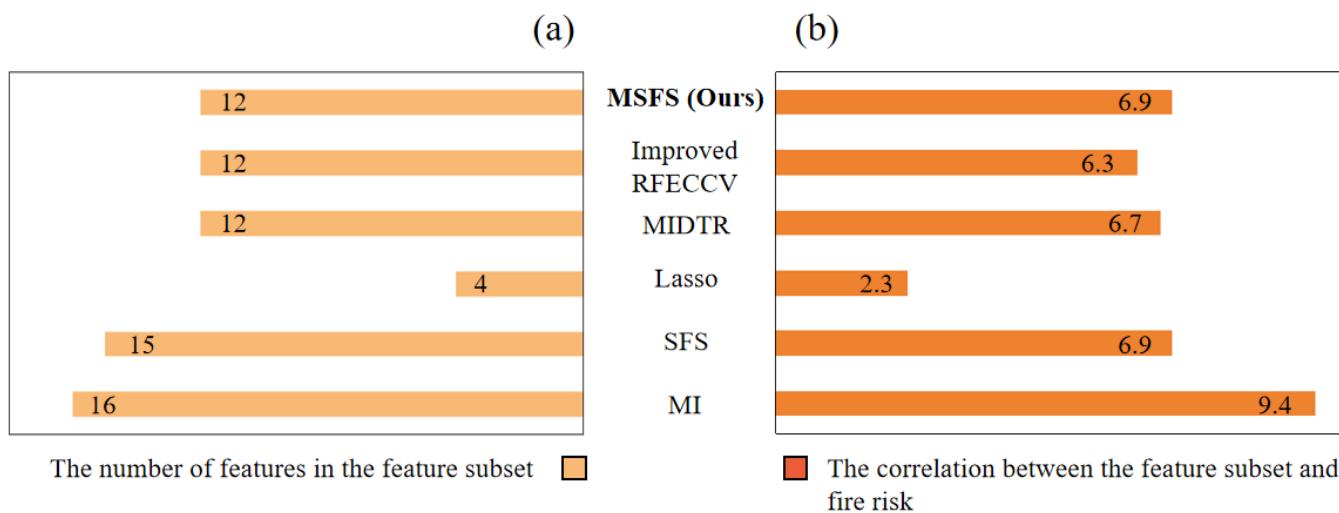


Figure 5. The statistical data obtained from various FS algorithms. (a) Number of features of different feature sets; (b) sum of correlation between distinct feature subsets and fire risk.

The contribution of feature subsets includes their correlation with fire risk and their importance to the model. We quantified the correlation between individual features and fire risk using the MI method and calculated the correlation of six feature subsets with fire risk (Figure 5b). This comparison allowed us to assess the ability of different FS algorithms to capture the correlation between features and fire risk. The feature subset derived from the MI algorithm exhibits the strongest correlation with fire risk, attributed to the large number of features selected. The MSFS and SFS algorithms are tied for second place, but the feature subset of the SFS algorithm contains three more features than that of the MSFS algorithm. Although the MIDTR and ImprovedRFECV algorithms selected the same number of features as the MSFS algorithm, their performance in correlating the subsets with fire risk lags by 2.89% and 8.70%, respectively. In conclusion, the MSFS algorithm stands out for its ability to maintain a high correlation between selected features and fire risk, even with a reduced feature set, highlighting its superiority in feature selection.

To assess the importance of feature subsets obtained from six FS algorithms, we plotted an importance graph for these subsets (Figure 6). The subsequent ten experiments based on AutoSklearn indicated that the best-performing model trained on different feature subsets was consistently the random forest (RF). Therefore, we visualized the importance of the original feature set using RF to represent the importance of the feature subsets. The results indicate that the feature subsets obtained by MI, SFS, ImprovedRFECV, and MSFS exhibit relatively high importance, successfully identifying most of the features significantly related to fire risk. The high importance of the MI and SFS feature subsets is attributed to the large number of features. In contrast, the feature subsets of MIDTR and Lasso show relatively low importance. Lasso's low importance is due to the small number of features in its subset, though it still retained the top two most important features. However, MIDTR eliminated most of the highly important features, including the most crucial feature, 'PRECmax'. Notably, despite having a similar number of features, the importance of MIDTR's subset is significantly lower than that of ImprovedRFECV and MSFS. Despite being 1.53% lower in importance compared to ImprovedRFECV, MSFS's feature subset exhibits a correlation that is 8.70% higher than that of ImprovedRFECV.

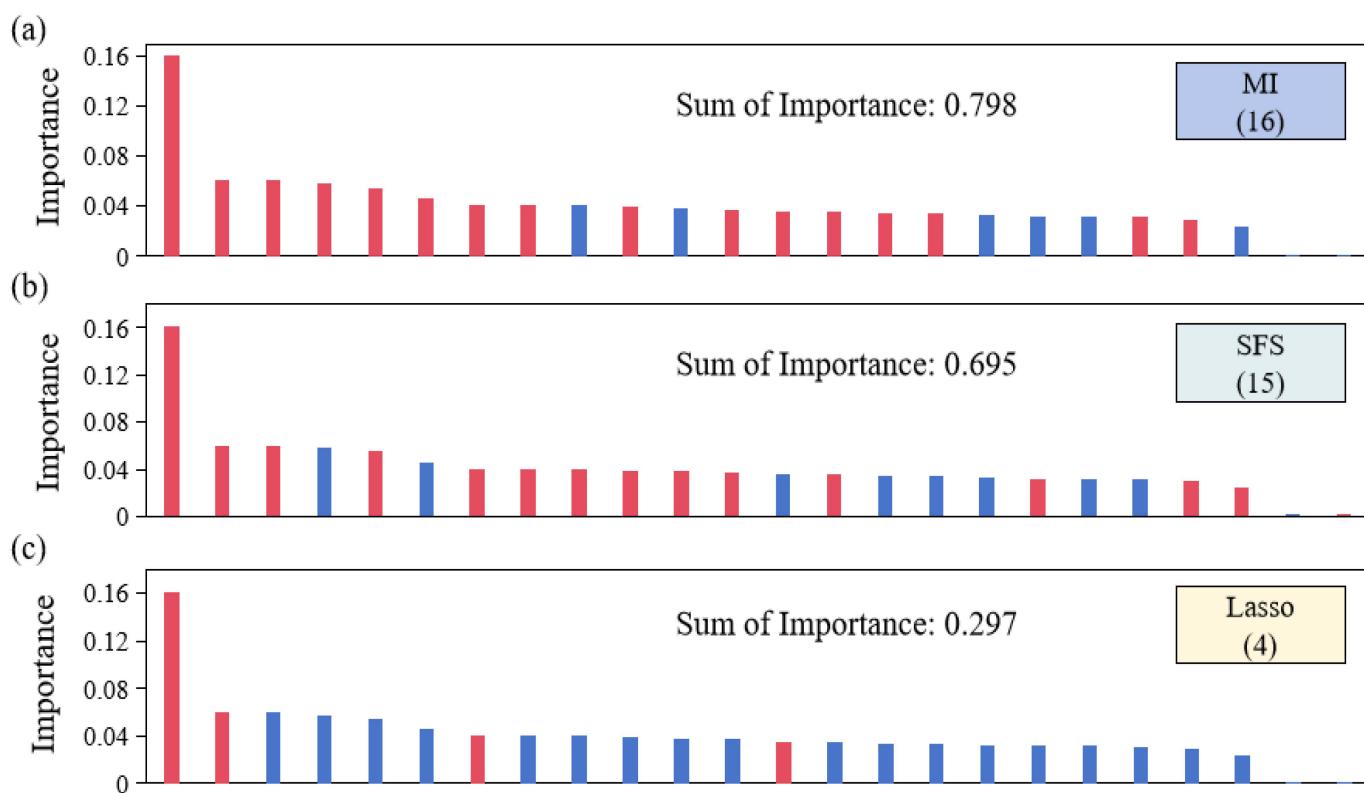


Figure 6. Cont.

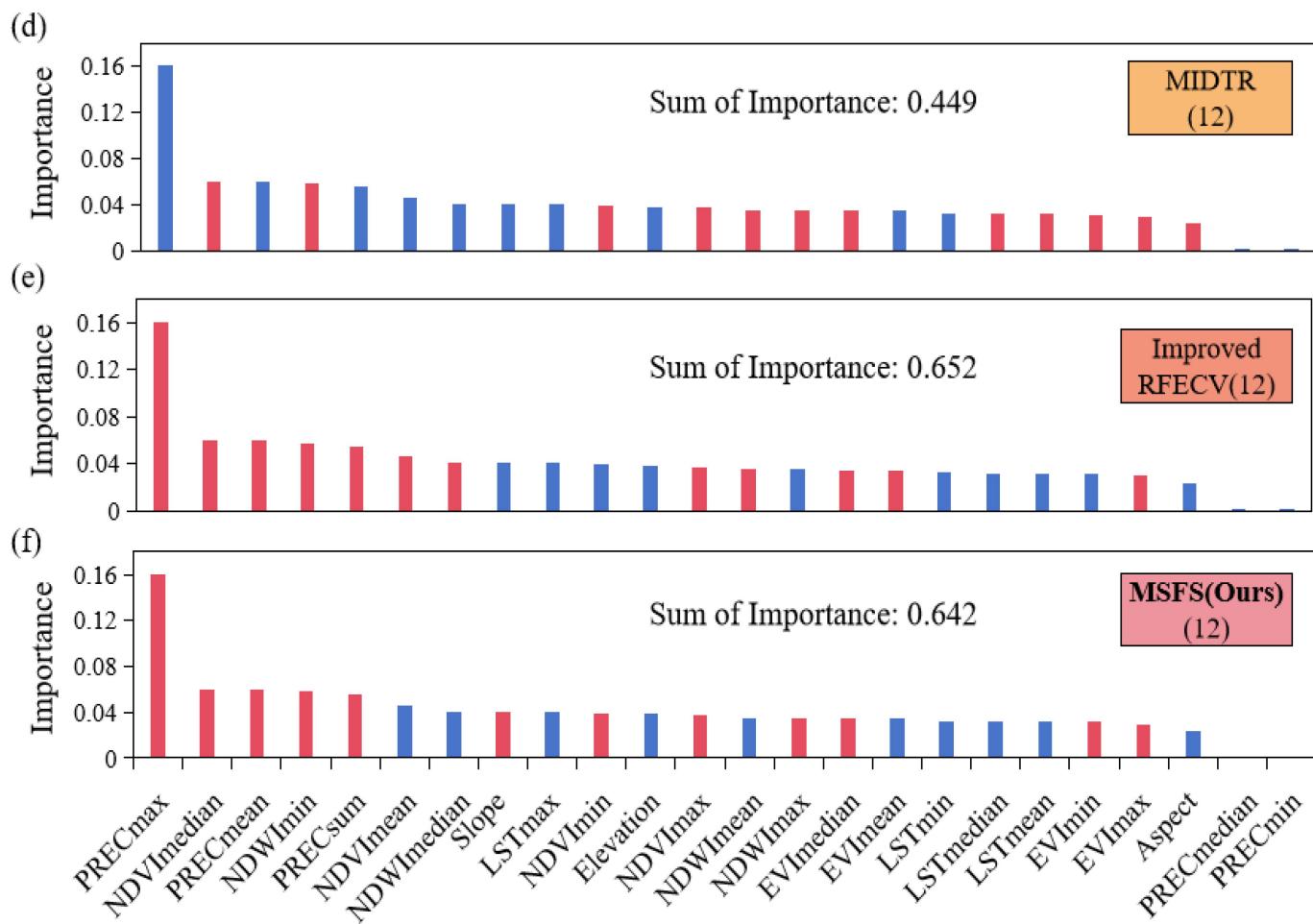


Figure 6. Importance of feature subsets for six FS algorithms. (a) MI; (b) SFS; (c) Lasso; (d) MIDTR; (e) ImprovedRFECV; and (f) MSFS. The red bars in the figure represent the subset of features selected by the corresponding algorithms, and the 24 features are ranked based on the feature importance of the RF algorithm. The numbers in parentheses of the FS algorithm represent the number of features.

Summing up the above, we concluded that for feature subsets with an equal number of features, the MSFS feature subset stands out as the highest contributor among the six evaluated FS algorithms.

4.1.2. Capability of FS Algorithms on Handling Collinear Features

Due to the feature extraction method of the dynamic environmental factors (Section 3.1), there is a significant amount of strong collinearity among the features in the forest fire feature set. Specifically, we define a correlation greater than 0.85 as indicating a strong collinear relationship. Figure 7 illustrates this phenomenon, revealing strong collinear relationships among features belonging to the EVI, NDVI, NDWI, and LST (e.g., EVI_{max}, EVI_{mean}, and EVI_{median}; NDVI_{max}, NDVI_{mean}, and NDVI_{median}; NDWI_{max}, NDWI_{mean}, and NDWI_{median}; LST_{max}, LST_{mean}, LST_{median}, and LST_{min}), while PREC features do not exhibit such collinearity. Notably, among the 16 features of the four indicators—the EVI, NDVI, NDWI, and LST—there exist 13 groups of strongly collinear features, as depicted in Figure 7.

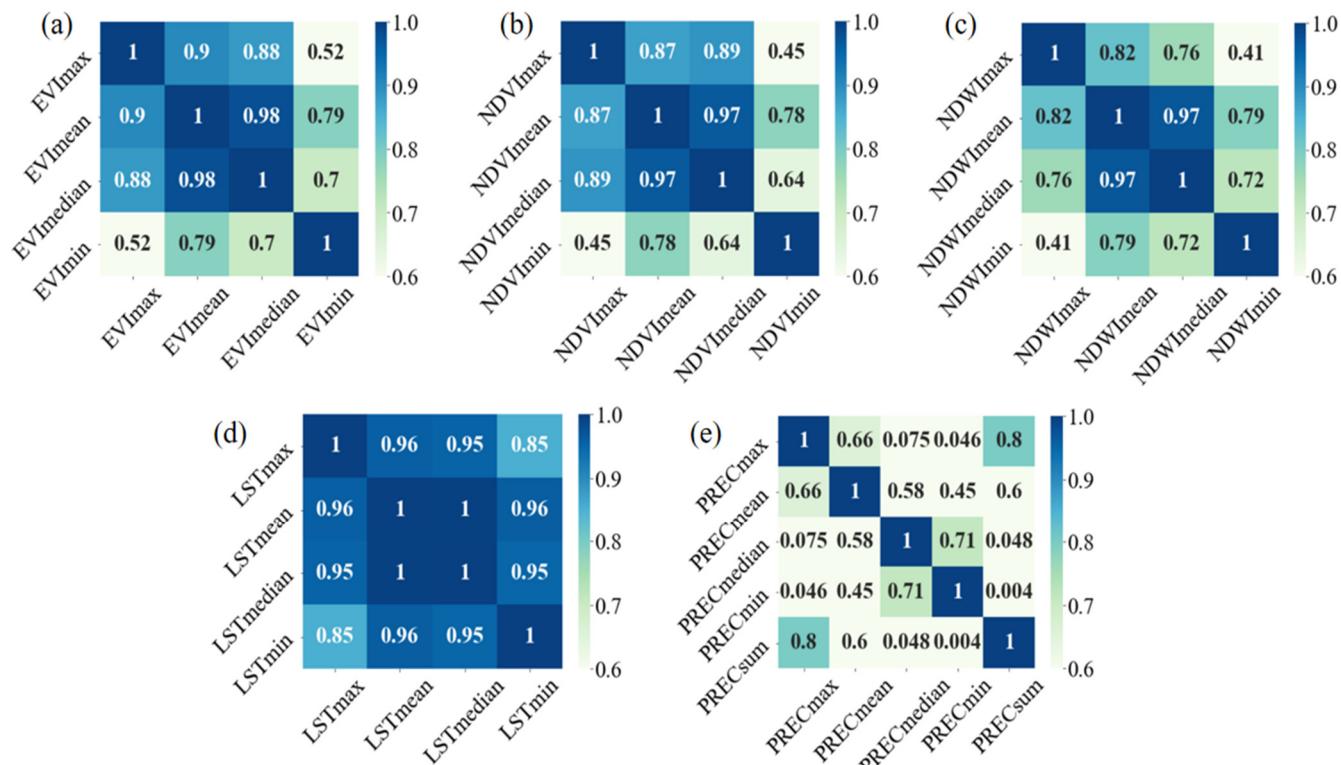


Figure 7. Heat map of correlation of environmental factors characterizing forest fires. (a) EVI; (b) NDVI; (c) NDWI; (d) LST; and (e) PREC.

Table 4 summarizes the collinear features and the number of collinear groups among the four dynamic indicators—the EVI, NDVI, NDWI, and LST—that remained in the feature subsets obtained from the six FS algorithms. The MSFS algorithm stood out by eliminating 11 groups of strongly collinear features, resulting in only two remaining groups in the selected subset. In contrast, the ImprovedRFECV, MIDTR, MI, SFS, and Lasso algorithms removed 8, 10, 5, 10, and 12 groups of strongly collinear features, respectively. Notably, Lasso retained the smallest number of collinear feature groups due to its subset containing only four features. Additionally, we found that the slightly higher importance of the ImprovedRFECV feature subset compared to that of MSFS could be attributed to the higher number of collinear features within the ImprovedRFECV subset. These collinear features provided similar importance, and their individual importance is relatively high. In conclusion, the proposed MSFS algorithm effectively considers the interrelationships between features and can handle collinearity issues among features well.

Table 4. Statistics of feature subsets among EVI, NDVI, NDWI, and LST indicators.

Algorithm	Feature Number of Subset	Number of Collinear Feature Groups	Specific Remaining Features of Each Indicator
MI	16	8	<i>EVI_{Max}, EVI_{Mean}, EVI_{Median}, EVI_{Min}</i> <i>NDVI_{Max}, NDVI_{Mean}, NDVI_{Median}, NDVI_{Min}</i> <i>NDWI_{Mean}, NDWI_{Median}, NDWI_{Max}, NDWI_{Min}</i>
SFS	15	3	<i>EVI_{Max}, EVI_{Median}</i> <i>NDVI_{Max}, NDVI_{Median}, NDVI_{Min}</i> <i>NDWI_{Max}, NDWI_{Median}</i> <i>LST_{Max}</i>
Lasso	4	1	<i>NDVI_{Mean}</i> <i>NDWI_{Mean}, NDWI_{Median}</i>

Table 4. Cont.

Algorithm	Feature Number of Subset	Number of Collinear Feature Groups	Specific Remaining Features of Each Indicator
MIDTR	12	3	<i>EVI_{max}, EVI_{median}, EVI_{min}</i> <i>NDVI_{max}, NDVI_{median}, NDVI_{min}</i> <i>NDWI_{max}, NDWI_{mean}, NDWI_{min}</i> <i>LST_{min}, LST_{median}</i>
ImprovedRFECV	12	5	<i>EVI_{median}, EVI_{mean}</i> <i>NDVI_{max}, NDVI_{median}, NDVI_{mean}</i> <i>NDWI_{median}, NDWI_{mean}, NDWI_{min}</i>
MSFS (Ours)	12	2	<i>EVI_{max}, EVI_{median}, EVI_{min}</i> <i>NDVI_{max}, NDVI_{median}, NDVI_{min}</i> <i>NDWI_{max}, NDWI_{min}</i>

Note: The features that have strong collinearity with other features are in bold italic, and the features that have strong multicollinearity are further underlined.

4.2. Model Prediction Performance Based on AutoSklearn and FS Algorithms

4.2.1. AutoSklearn Training Process Based on Different Feature Sets

Table S1 (see Supplementary Materials) summarizes the training process of AutoSklearn over ten experiments. The analysis shows that, overall, after feature selection, AutoSklearn selects a more diverse set of models, successfully runs a greater number of models, and experiences fewer instances of model runtime exceeding the time limit. Additionally, in the fourth, fifth, and eighth experiments, AutoSklearn without feature selection (non-FS) encountered model execution failures, while most instances with feature selection did not experience such issues. This indicated that with the optimization of feature selection, AutoSklearn expanded its model selection and search abilities, thereby enhancing its overall efficiency. Table 5 reveals that among the six FS algorithms, the SFS algorithm attained the highest validation score (using accuracy as the validation metric), followed closely by the MSFS algorithm. However, the SFS algorithm encountered model execution crashes in the ninth and tenth experiments, whereas the MSFS algorithm remained unscathed in all ten experiments. This validates that the introduction of the MSFS algorithm more effectively improves the operational efficiency of AutoSklearn. Moreover, the best validation scores across the ten experiments for the MSFS, SFS, and Lasso algorithms were identical, demonstrating the stability of their feature subsets during the AutoSklearn training process.

Table 5. The best validation scores of AutoSklearn based on six FS algorithms.

Experiment	MI	SFS	Lasso	MIDTR	ImprovedRFECV	MSFS (Ours)
1	0.906	0.918	0.884	0.869	0.906	0.909
2	0.906	0.918	0.884	0.869	0.906	0.909
3	0.906	0.918	0.884	0.869	0.906	0.909
4	0.906	0.918	0.884	0.869	0.906	0.909
5	0.906	0.918	0.884	0.869	0.906	0.909
6	0.906	0.918	0.884	0.869	0.906	0.909
7	0.908	0.918	0.884	0.862	0.904	0.909
8	0.908	0.918	0.884	0.869	0.904	0.909
9	0.907	0.918 ×	0.884	0.864	0.905	0.909
10	0.908	0.918 ×	0.884	0.869	0.904	0.909

Note: Bold and red indicate first place; bold and blue indicate second place; × indicates a crash in the model runs. The best validation score during the training process is evaluated using accuracy.

Table 6 lists the model success rates of ten experiments based on seven feature subsets, comparing the ability of different FS algorithms to enhance model operational efficiency and stability. The data indicated that in most of the ten experiments, the execution success

rates of AutoSklearn generally improved after feature selection. However, there were slight decreases in the success rates of ImprovedRFECV in the second experiment and SFS and MIDTR in the third experiment. This finding confirms that feature selection can increase the probability of successful model execution by AutoSklearn. Furthermore, AutoSklearn achieved the most significant increase in model execution success rate after applying Lasso selection, with an average improvement of approximately 27%. This improvement is attributed to the reduced number of features selected by Lasso, which accelerates the model execution speed and alleviates the burden of high-dimensional data on model performance. When comparing MIDTR and ImprovedRFECV, which have the same number of features as MSFS, the models selected by MSFS exhibited an average increase in execution success rate of about 15%, which is approximately 5% and 7% higher than MIDTR and ImprovedRFECV, respectively. This underscores the superior effectiveness of MSFS feature subsets. In conclusion, the fewer the features, the higher the probability of successful model operation. Additionally, for feature subsets with the same number of features, the success probability of the models is influenced by the effectiveness of the feature subset.

Table 6. Success rate of model runs (%).

Experiment	Non-FS (24)	MI (16)	SFS (15)	Lasso (4)	MIDTR (12)	ImprovedRFECV (12)	MSFS (ours) (12)
1	63	68 ↑	81 ↑	85 ↑	75 ↑	71 ↑	78 ↑
2	62	66 ↑	70 ↑	84 ↑	68 ↑	38 ↓	71 ↑
3	63	63	42 ↓	88 ↑	59 ↓	67 ↑	68 ↑
4	64	73 ↑	68 ↑	86 ↑	75 ↑	71 ↑	78 ↑
5	64	66 ↑	74 ↑	87 ↑	71 ↑	76 ↑	77 ↑
6	54	66 ↑	70 ↑	87 ↑	73 ↑	71 ↑	71 ↑
7	48	69 ↑	71 ↑	88 ↑	61 ↑	68 ↑	71 ↑
8	52	66 ↑	70 ↑	87 ↑	68 ↑	70 ↑	71 ↑
9	63	76 ↑	80 ↑	88 ↑	71 ↑	71 ↑	79 ↑
10	63	73 ↑	80 ↑	88 ↑	75 ↑	75 ↑	78 ↑
Average Improvement	59.60	68.60 ↑ 9.00	70.60 ↑ 11.00	86.80 ↑ 27.20	69.60 ↑ 10.00	67.80 ↑ 8.20	74.56 ↑ 14.96

Note: numbers below algorithms represent the number of features in the corresponding feature set. ↓ indicates a decrease in the success rate of model runs, and ↑ indicates an increase. Numbers in red and bold indicate first place, while blue and bold indicate second place.

4.2.2. Prediction Performance of Different Feature Sets Based on AutoSklearn

Table 7 presents the average values of model prediction performance, evaluated based on ten experiments with seven feature subsets and AutoSklearn (darker red indicates better performance and darker blue indicates worse performance in each column). It can be seen that, although the MSFS algorithm ranked second in the best validation score during the AutoSklearn training process (Table 6), its performance on the test set was superior to the SFS algorithm across the average values of all five evaluation metrics. Additionally, although Lasso has a higher ability to remove collinear features compared to MSFS, the limited information provided by fewer features lowers the model's predictive capabilities. Specifically, the accuracy, precision, recall, F1-score, and the ROC_AUC of the MSFS-based model are approximately 3%, 5%, 4%, 5%, and 4% higher than those of the Lasso-based model, respectively. Compared to non-FS, MSFS demonstrates a marginal 0.2% decrease in the recall, while the other metrics are all higher, especially precision, which is improved by approximately 2%. In the context of forest fire risk prediction, improved precision helps reduce false positives, thereby preventing the ineffective deployment and waste of firefighting resources. This study also opted for the F1-score as a holistic metric to balance the model's precision and recall capabilities in predicting positive samples (i.e., FPs). In forest fire risk prediction, the model's proficiency lies not only in accurately predicting fires (i.e., high precision) but also in identifying as many real fire events as possible (i.e.,

high recall). Despite the slight decrease of 0.2% in recall, the F1-score for MSFS improved by approximately 1%, signifying its better performance in a comprehensive evaluation. Furthermore, the number of features in the MSFS subset is just half that of non-FS. Thus, this study validates that integrating the MSFS algorithm into AutoSklearn not only enhances its operational efficiency but also bolsters its prediction performance.

Table 7. Average prediction performance of ten experiments over seven feature sets.

Algorithm	Evaluation Metrics				
	Accuracy	Precision	Recall	F1-Score	ROC_AUC
Non-FS	0.927	0.889	0.861	0.875	0.908
MI	0.923	0.885	0.852	0.868	0.903
SFS	0.927	0.893	0.857	0.875	0.907
Lasso	0.904	0.856	0.814	0.835	0.878
MIDTR	0.891	0.896	0.715	0.795	0.842
ImprovedRFECV	0.924	0.880	0.860	0.870	0.905
MSFS (ours)	0.931	0.903	0.859	0.881	0.910

Low

High

Note: Darker red indicates better performance and darker blue indicates worse performance in each column. Bolded numbers indicate the highest scores in each column.

4.2.3. Stability Analysis of FS Algorithms for AutoSklearn Modeling

To validate the ability of the MSFS algorithm to enhance the stability of model prediction performance, we plotted box plots to visualize five prediction performance metrics across ten experiments (Figure 8). From the plots, we observed that the box sizes for non-FS, MI, SFS, and ImprovedRFECV are relatively small, while the box plot for MSFS appears almost as a single line. This typically indicates a high concentration of MSFS experimental results with minimal variation, suggesting stable performance. The minimal variance in MSFS results demonstrates it is less sensitive to variations in input data, making it more reliable and adaptable to dynamically changing forest environments. Conversely, the larger box sizes for MIDTR and Lasso indicate a wide range of experimental results, suggesting these algorithms are unstable and the feature subsets obtained are not robust. This also addresses the question raised in the introduction about the unknown stability performance of the MIDTR algorithm.

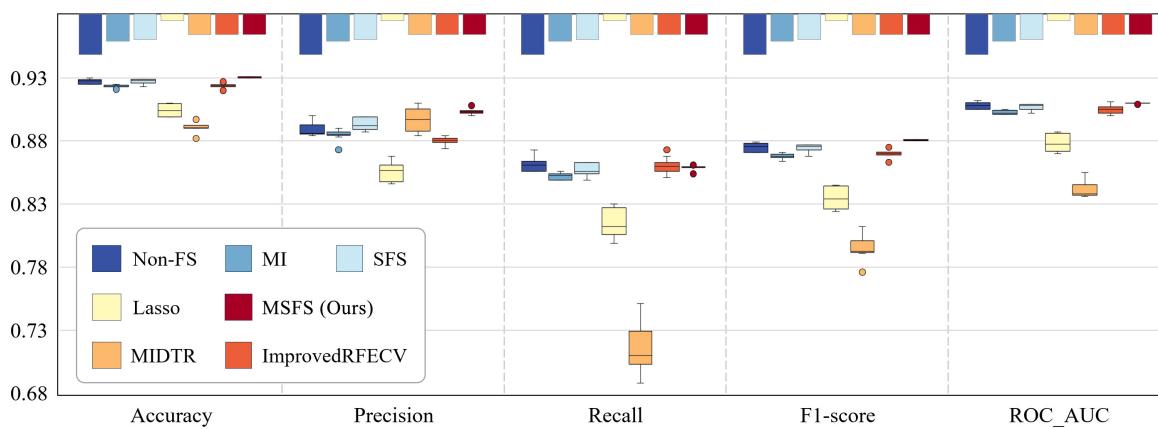


Figure 8. Box plots of five prediction performance metrics for non-FS and six FS algorithms across ten experiments. The bar charts at the top visually display the number of features in the corresponding feature subsets.

To quantitatively assess the stability of FS algorithms, we calculated the box height, defined as the difference between the upper and lower quartiles, as a metric for measuring the performance variation in feature subsets. Since this study just involved 10 scattered experimental results, the quartile positions were obtained using interpolation. Table 8

presents the performance variation in the feature subsets based on ten experiments. The MSFS feature subset exhibits the lowest performance variation across the five evaluation metrics, with an average variation of 0.09%. This variation is significantly lower than that of non-FS, MI, SFS, Lasso, MIDTR, and ImprovedRFECV, specifically by 0.54%, 0.20%, 0.51%, 1.68%, 1.18%, and 0.29%, respectively. This verifies the superior stability of the MSFS algorithm and its ability to generate feature subsets with minimal performance variation.

Table 8. Performance variations in seven feature subsets based on ten experiments (%).

Algorithm	Evaluation Metrics					Average
	Accuracy	Precision	Recall	F1-Score	ROC_AUC	
Non-FS	0.35	0.73	0.80	0.72	0.55	0.63
MI	0.12	0.25	0.50	0.25	0.32	0.29
SFS	0.30	1.00	0.90	0.40	0.40	0.60
Lasso	1.10	1.67	2.57	1.95	1.55	1.77
MIDTR	0.25	1.78	2.62	0.90	0.82	1.27
ImprovedRFECV	0.15	0.35	0.68	0.22	0.50	0.38
MSFS (ours)	0.10	0.20	0.05	0.10	0.00	0.09

To further visually demonstrate the stability of the MSFS algorithm, we plotted a line graph of the ten experimental results (excluding MIDTR and Lasso due to their poor performance) (Figure 9). This graph provides a comparative analysis of the stability performance of the MSFS algorithm with non-FS, MI, SFS, and ImprovedRFECV. We observed minimal fluctuations in the ten experimental results of MSFS, particularly in Figure 9d, showing the ten F1-scores, and Figure 9e, displaying the ten ROC_AUC scores, which are almost straight lines. This indicates that the MSFS algorithm exhibits a high stability and reliability in predicting forest fire risk. In contrast, non-FS and other FS algorithms exhibited significant fluctuations. Although the recall of non-FS was slightly higher than MSFS, we found considerable fluctuations in the ten recall results of non-FS. Specifically, the recall scores of the third and seventh experiments were significantly high, approximately 2% higher than the other experiments, suggesting a low confidence level in these two results. This also inflated the average recall score of non-FS over ten experiments, resulting in a slightly higher value than MSFS.

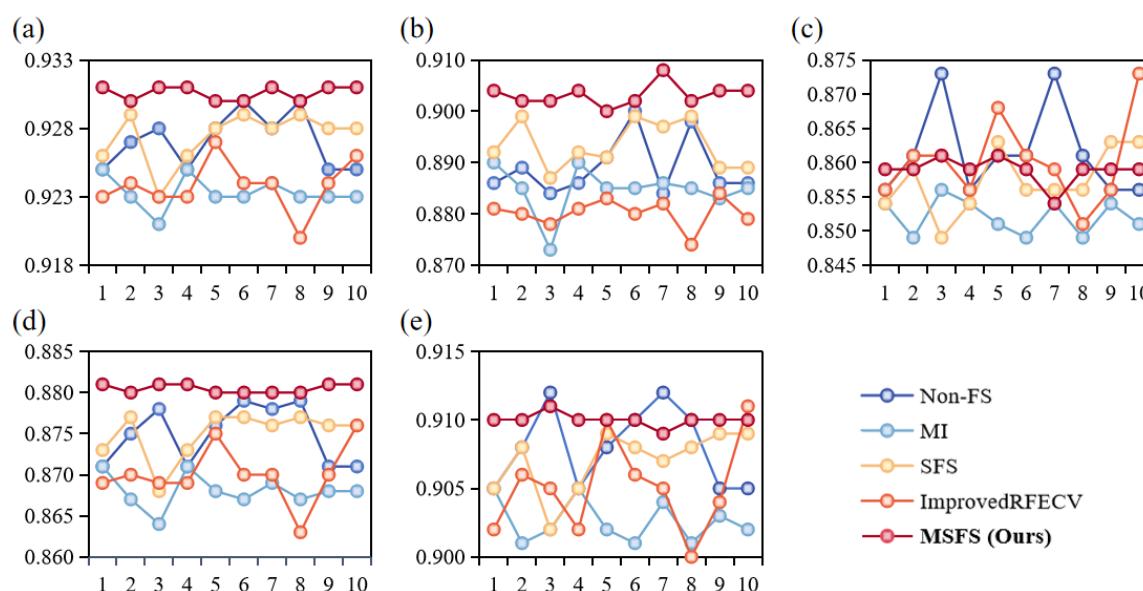


Figure 9. Stability analysis of five evaluation metrics over non-FS, MI, SFS, ImprovedRFECV, and MSFS. (a) Accuracy; (b) precision; (c) recall; (d) F1-score; and (e) ROC_AUC.

In conclusion, the forest fire risk prediction model constructed using the feature subsets obtained from the MSFS algorithm exhibits a high predictive stability, demonstrating that the proposed MSFS algorithm can obtain feature sets with significant contributions and high stability.

5. Discussion

5.1. The Reproducible Challenge of AutoSklearn and Advantages of the MSFS Algorithm

Influenced by factors such as hardware differences (e.g., CPU, GPU models, and performance), operating system versions, and discrepancies in the implementation of dependency libraries, the experimental results of AutoML are difficult to reproduce completely, with AutoSklearn being no exception [58]. Therefore, when aiming for reproducibility, it is necessary to comprehensively take these factors into account and perform testing and validation in similar environments to the utmost extent possible. Despite this, due to the inherent randomness of AutoSklearn, there may still be minor differences between results. For example, upon inspecting the training process of AutoSklearn, it was found that the number of models selected, the number of successfully executed models, and the number of models encountering timeouts varied in most experiments (refer to Supplementary Materials Table S1 for details). This study conducted ten experiments, each using fivefold cross-validation, equivalent to 50 experiments, to investigate the variability of AutoSklearn's experimental results and the stability of the MSFS algorithm. Notably, this study successfully replicated the results of AutoSklearn's first experiment in the fourth experiment, as detailed in Table S2 (see Supplementary Materials).

This multi-stage feature selection algorithm, MSFS, proposed in this study combines the advantages of MIG and RFECV, compensating for their respective shortcomings. It can consider both the strong correlation with the target variable and the interrelationships among features. However, MIG calculations may be subject to randomness, especially during dataset sampling or feature selection, making it challenging to ensure stability, thus affecting the performance of feature subsets [59,60]. Similarly, the RFECV algorithm lacks stability evaluation methods, leading to potential uncertainties and biases during random sampling and perturbations [32]. Its results may be influenced by specific datasets and cross-validation folds, resulting in feature subsets that lack robustness. To address these stability issues, the MSFS algorithm executes a voting aggregation mechanism after the first and second screening stages. By selecting features that are consistently chosen in over half of the iterations, this mechanism ensures that the resulting feature subsets have a high stability.

Through the comprehensive analysis of the results in Section 4, this study validated the MSFS feature subset possesses a high contribution, a remarkable collinearity handling ability, and robust algorithmic stability. The research also confirmed that the introduction of MSFS into AutoSklearn significantly improves its performance, offering a broader range of model selections and a higher probability of successful model training, thereby enhancing the operational efficiency of AutoSklearn. Additionally, due to the high environmental configuration requirements and inherent randomness of AutoSklearn, a stable feature set becomes paramount. Even in situations where reproducibility is challenging, a stable feature set can significantly reduce the variability of experimental results, thereby enhancing the reliability and confidence of the outcomes.

5.2. The Value of MSFS–AutoSklearn in the Application of Efficient Task Implementation

To better elucidate the significance and value of integrating MSFS with AutoSklearn for diverse application tasks, we summarized and categorized five stages of applying ML to real-world applications, as illustrated in Figure 10. We first defined a set of levels to represent varying degrees of automation offered by existing systems. Lower levels signify less automation and more manual effort, while higher levels imply more automation and less manual involvement.

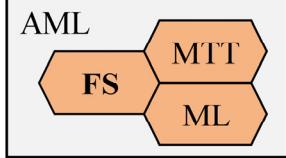
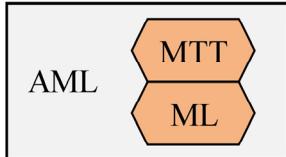
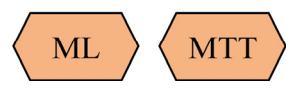
Level	Systems	Automation Degree	Application Personnel Productivity
Level 5	MSFS-AutoSklearn		
Level 4	AutoML		
Level 3	Scikit-Learn, Pytorch, Keras, Tensorflow...		
Level 2	Basic Implementation of Random Forest, SVM, MLP...		
Level 1	Programming Languages Like Python, R, Java...		

Figure 10. Automation levels for end-to-end ML workflows (modified from [6]). The figure illustrates examples of existing systems at each level, accompanied by a color gradient indicating the productivity of application personnel lacking ML experience. As shown, advancing to higher automation levels yields two main benefits. Firstly, it enables application personnel to leverage ML. Secondly, it amplifies the productivity of application personnel by reducing their manual workload as the automation level increases. ML, machine learning; MTT, models training and testing; AML, automated machine learning.

Level 1: no Automation. All ML tasks require extensive manual coding lacking any automation support. For example, developing a new prediction model necessitates personnel to construct the core ML model from scratch using a chosen programming language. This process is time-consuming and hinders the widespread adoption and application of ML.

Level 2: only ML automated. Level 2 offers the basic implementations of core ML models, marking the lowest level of automation a system can offer. For instance, the implementation of a SVM classifier in Python serves as a library for application personnel, but tasks such as model selection, training, and hyperparameter tuning still require manual execution, limiting the level of automation.

Level 3: ML + MTT automated separately. Open-source libraries like Scikit-Learn, PyTorch, and TensorFlow come with a rich set of ML models and algorithms that application personnel can directly call upon without the need for intricate coding. However, crucial decisions such as model selection and hyperparameter tuning still rely on the professional judgment of personnel to optimize model performance and precise task execution.

Level 4: ML + MTT automated jointly. AutoML incorporates multiple ML models, automatically selecting, training, and tuning models to lower the barrier to ML usage. Yet, most open-source AutoML solutions lack sufficient research on automated feature selection, primarily applicable to low-dimensional data. When dealing with high-dimensional data, the manual or guided selection of a FS algorithm is still necessary, emphasizing the significance of the effectiveness and stability of FS algorithms for AutoML.

Level 5: FS + ML + MTT automated jointly. Combing MSFS with AutoSklearn effectively addresses the shortcomings of AutoML in automated feature selection. The well-performing MSFS algorithm enables AutoML to handle high-dimensional data, improve prediction reliability, expand model choices, speed up training, and elevate training success rates. This frees up application personnel to focus on business logic and problem-solving, improving efficiency and solution quality. Additionally, the combined framework adapts to dynamic data changes, optimizing models for new data distributions and patterns, making it ideal for handing real-time data streams and dynamic scenarios.

5.3. MSFS–AutoSklearn Holds Broad Prospects in Efficient Multi-Task Applications

The MSFS–AutoSklearn framework breaks down the barriers between specialized technology and practical applications by lowering technical thresholds, increasing efficiency, accelerating model deployment, reducing talent demands, and fostering innovation. This framework promotes the widespread application and adoption of ML technology across various fields, driving innovation and development in related industries. Part of the application fields are as follows:

In the field of remote sensing, the combination of AutoSklearn and the MSFS algorithm will bring revolutionary advancements. This approach can extract the most relevant features from spectral, textural, shape, and multi-scale data derived from remote sensing, significantly improving data processing efficiency and model accuracy. Additionally, it reduces the need for expert intervention and offers excellent scalability. These improvements will provide timely and effective decision support for agriculture, forestry, urban planning, environment monitoring, and disaster management, promoting the widespread application and continuous innovation of remote sensing technology.

In the field of recommender systems, user behavior data (such as purchase history, browsing records, etc.) and item attribute data (such as product categories, prices, reviews, etc.) can serve as features, collectively forming high-dimensional user-item interaction data. Given the widespread application of AutoML in recommender systems, the MSFS–AutoSklearn framework is expected to further enhance the accuracy, efficiency, and personalization of recommendations. This improvement will better meet user needs and significantly enhance user experience.

In the fields of bioinformatics and genomics, the application of the MSFS–AutoSklearn framework can automatically extract and effectively select key features such as gene expression, gene variation, protein structure, and function. This enables researchers to analyze complex biological data more accurately and efficiently, uncovering underlying biological mechanisms. This not only accelerates disease diagnosis, prediction, and new drug development, but also significantly enhances the effectiveness of personalized medicine, driving continuous innovation and progress in the biomedical field.

In the fields of finance, predicting stock markets and analyzing trading data involve numerous data dimensions. The application of MSFS–AutoSklearn simplifies and enhances the efficiency of complex financial data analysis. Through automated feature selection and model training, financial institutions can more accurately forecast market trends, optimize investment strategies, and reduce investment risks. Additionally, AutoSklearn facilitates the rapid deployment of novel ML models, allowing institutions to adapt to the ever-changing financial market environment and foster continuous innovation, enhancing customer experience and market competitiveness.

6. Conclusions

To address the challenges faced by AutoML in handling large-scale data, this study proposes a multi-stage feature selection algorithm, the MSFS algorithm, and integrates it with AutoSklearn to construct an efficient task implementation modeling framework, MSFS–AutoSklearn, with forest fire risk prediction as an application case. The MSFS algorithm comprises three stages: MIG, RFECV, and a voting aggregation mechanism, allowing the algorithm to comprehensively consider the correlations among features, the correlations

between features and the target variable, and the stability of feature subsets. The study validates the excellent dimensionality reduction capability of MSFS, retaining only half of the features, and significantly enhancing the ability of AutoSklearn to handle high-dimensional data tasks. Through a comparison with five FS algorithms—MI, SFS, Lasso, MIDTR, and ImprovedRFECV—the proposed MSFS algorithm demonstrates a higher feature subset contribution and excellent collinearity handling capabilities. The MSFS–AutoSklearn framework achieves richer model selection, higher success rates in model training, superior prediction stability, and optimal prediction performance. In summary, the MSFS–AutoSklearn framework not only improves the efficiency and productivity of prediction tasks but also enhances prediction accuracy, showing significant potential for efficient implementation in various real-world tasks. This research provides strong support for the widespread application of AutoML.

Future research directions can be expanded based on the findings of this study. For instance, it is possible to explore the application of the MSFS–AutoSklearn framework in other fields (such as healthcare, finance, etc.) to verify its effectiveness under different data characteristics and structures. At the same time, developing more advanced algorithms for the automation and intelligence of feature selection within the MSFS–AutoSklearn framework to cope with increasingly complex data environments will be an important direction for future research.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/rs16173190/s1>, Table S1: AutoSklearn training process of ten experiments; Table S2: Prediction performance based on seven feature sets of ten experiments.

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

Funding: This research was funded by the National Key Research and Development Program of China (2021YFF0703901), the National Natural Science Foundation of China (42171323, 42271353), the Guangdong Basic and Applied Basic Research Foundation (2023A1515011261, 2024A1515011858), the Undertaking National Science and Technology Major Project by Shenzhen Technology and Innovation Bureau (CJGJZD20220517141800002), and the Excellent Youth Innovation Foundation of the Shenzhen Institute of Advanced Technology of the Chinese Academy of Science (E3G044).

Data Availability Statement: All the data used in this study are available publicly. The VIIRS fire data are available from NASA FIRMS (https://firms.modaps.eosdis.nasa.gov/active_fire/ (accessed on 27 February 2024)). The NASADEM data is available from USGS (https://lpdaac.usgs.gov/products/nasadem_hgtv001/ (accessed on 27 February 2024)). The CLCD data are available at <https://doi.org/10.5281/zenodo.4417810> (accessed on 27 February 2024). The CHIRPS data are available from the Climate Hazards Center of University of California, Santa Barbara (UCSB/CHG) (<https://data.chc.ucsb.edu/products/CHIRPS-2.0/> (accessed on 27 February 2024)). The MYD11A1 data is available from USGS (<https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/MYD11A1> (accessed on 27 February 2024)). The MYD09GA data is available from USGS (<https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/MYD09GA> (accessed on 27 February 2024)).

Conflicts of Interest: Author Yuankai Ge was employed by the company The Eighth Engineering Co., Ltd., China Tiesiju Civil Engineering Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

- Huber, M.; Zller, M. Benchmark and Survey of Automated Machine Learning Frameworks. *Artif. Intell. Res.* **2021**, *70*, 409–472. [\[CrossRef\]](#)
- Guyon, I.; Chaabane, I.; Escalante, H.J.; Escalera, S.; Jajetic, D.; Lloyd, J.R.; Macià, N.; Ray, B.; Romaszko, L.; Sebag, M.; et al. A brief Review of the ChaLearn AutoML Challenge: Any-time Any-dataset Learning without Human Intervention. In Proceedings of the Workshop on Automatic Machine Learning; PMLR: New York, NY, USA, 2016; pp. 21–30.
- Alsharef, A.; Aggarwal, K.; Sonia; Kumar, M.; Mishra, A. Review of ML and AutoML Solutions to Forecast Time-Series Data. *Arch. Comput. Methods Eng.* **2022**, *29*, 5297–5311. [\[CrossRef\]](#) [\[PubMed\]](#)
- Shen, Z.; Zhang, Y.; Wei, L.; Zhao, H.; Yao, Q. Automated Machine Learning: From Principles to Practices. *arXiv* **2018**. [\[CrossRef\]](#)
- Wever, M.; Tornede, A.; Mohr, F.; Hullermeier, E. AutoML for Multi-Label Classification: Overview and Empirical Evaluation. *IEEE Trans. Pattern Anal.* **2021**, *43*, 3037–3054. [\[CrossRef\]](#)
- Karmaker Santu, S.K.; Hassan, M.M.; Smith, M.J.; Xu, L.; Zhai, C.; Veeramachaneni, K. AutoML to Date and Beyond: Challenges and Opportunities. *ACM Comput. Surv.* **2022**, *54*, 1–36. [\[CrossRef\]](#)
- Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
- Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [\[CrossRef\]](#) [\[PubMed\]](#)
- Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.; Blum, M.; Hutter, F. Efficient and Robust Automated Machine Learning. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2755–2763.
- Li, J.; Cheng, K.; Wang, S.; Morstatter, F.; Trevino, R.P.; Tang, J.; Liu, H. Feature Selection. *ACM Comput. Surv.* **2018**, *50*, 1–45. [\[CrossRef\]](#)
- Kumar, V.; Minz, S. Feature Selection: A literature Review. *Smart Comput. Rev.* **2014**, *4*, 211–229. [\[CrossRef\]](#)
- Jovic, A.; Brkic, K.; Bogunovic, N. A review of feature selection methods with applications. In Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 1200–1205.
- Dhal, P.; Azad, C. A comprehensive survey on feature selection in the various fields of machine learning. *Appl. Intell.* **2022**, *52*, 4543–4581. [\[CrossRef\]](#)
- Hancer, E.; Xue, B.; Zhang, M. Differential evolution for filter feature selection based on information theory and feature ranking. *Knowl.-Based Syst.* **2018**, *140*, 103–119. [\[CrossRef\]](#)
- Ruiz, R.; Riquelme, J.C.; Aguilar-Ruiz, J.S. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recogn.* **2006**, *39*, 2383–2392. [\[CrossRef\]](#)
- Chen, Y.; Bi, J.; Wang, J.Z. MILES: Multiple-instance learning via embedded instance selection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1931–1947. [\[CrossRef\]](#)
- Aram, K.Y.; Lam, S.S.; Khasawneh, M.T. Linear Cost-sensitive Max-margin Embedded Feature Selection for SVM. *Expert Syst. Appl.* **2022**, *197*, 116683. [\[CrossRef\]](#)
- Bommert, A.; Welchowski, T.; Schmid, M.; Rahnenführer, J. Benchmark of filter methods for feature selection in high-dimensional gene expression survival data. *Brief. Bioinform.* **2022**, *23*, bbab354. [\[CrossRef\]](#) [\[PubMed\]](#)
- Haq, A.U.; Zeb, A.; Lei, Z.; Zhang, D. Forecasting daily stock trend using multi-filter feature selection and deep learning. *Expert Syst. Appl.* **2021**, *168*, 114444. [\[CrossRef\]](#)
- Vergara, J.R.; Estévez, P.A. A review of feature selection methods based on mutual information. *Neural Comput. Appl.* **2014**, *24*, 175–186. [\[CrossRef\]](#)
- Tan, M.; Pu, J.; Zheng, B. Optimization of breast mass classification using sequential forward floating selection (SFFS) and a support vector machine (SVM) model. *Int. J. Comput. Ass. Rad.* **2014**, *9*, 1005–1020. [\[CrossRef\]](#)
- Liu, Y.; Zheng, Y.F. FS_SFS: A novel feature selection method for support vector machines. *Pattern Recogn.* **2006**, *39*, 1333–1345. [\[CrossRef\]](#)
- Monirul Kabir, M.; Monirul Islam, M.; Murase, K. A new wrapper feature selection approach using neural network. *Neurocomputing* **2010**, *73*, 3273–3283. [\[CrossRef\]](#)
- Maldonado, S.; Weber, R. A wrapper method for feature selection using Support Vector Machines. *Inform. Sci.* **2009**, *179*, 2208–2217. [\[CrossRef\]](#)
- Fonti, V.; Belitsker, E. Feature Selection using LASSO. In *Research Paper in Business Analytics*; Vrije Universiteit Amsterdam: Amsterdam, The Netherlands, 2017; pp. 1–25.
- Liu, H.; Zhou, M.C.; Liu, Q. An Embedded Feature Selection Method for Imbalanced Data Classification. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 703–715. [\[CrossRef\]](#)
- Yin, Y.; Jang-Jaccard, J.; Xu, W.; Singh, A.; Zhu, J.; Sabrina, F.; Kwak, J. IGRF-RFE: A Hybrid Feature Selection Method for MLP-based Network Intrusion Detection on UNSW-NB15 Dataset. *J. Big Data* **2023**, *10*, 15. [\[CrossRef\]](#)
- Omuya, E.O.; Okeyo, G.O.; Kimwele, M.W. Feature Selection for Classification using Principal Component Analysis and Information Gain. *Expert Syst. Appl.* **2021**, *174*, 114765. [\[CrossRef\]](#)

29. Bolón-Canedo, V.; Alonso-Betanzos, A. Ensembles for feature selection: A review and future trends. *Inform. Fusion* **2019**, *52*, 1–12. [[CrossRef](#)]
30. Khaire, U.M.; Dhanalakshmi, R. Stability of feature selection algorithm: A review. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 1060–1073. [[CrossRef](#)]
31. Tu, T.; Su, Y.; Ren, S. FC-MIDTR-WCCA: A Machine Learning Framework for PM2.5 Prediction. *IAENG Int. J. Comput. Sci.* **2024**, *51*, 544–552.
32. Tu, T.; Su, Y.; Tang, Y.; Tan, W.; Ren, S. A More Flexible and Robust Feature Selection Algorithm. *IEEE Access* **2023**, *11*, 141512–141522. [[CrossRef](#)]
33. Machado-Silva, F.; Libonati, R.; Lima, T.F.M.D.; De, M.; Dacamara, C.C. Drought and fires influence the respiratory diseases hospitalizations in the Amazon. *Ecol. Indic.* **2019**, *109*, 105817. [[CrossRef](#)]
34. Machado, A.; Serpa, D.; Santos, A.K.; Gomes, A.P.; Keizer, J.J.; Oliveira, B.R.F. Effects of different amendments on the quality of burnt eucalypt forest soils—A strategy for ecosystem rehabilitation. *J. Environ. Manag.* **2022**, *320*, 115766. [[CrossRef](#)]
35. Vasilakos, C.; Kalabokidis, K.; Hatzopoulos, J.; Matsinos, I. Identifying wildland fire ignition factors through sensitivity analysis of a neural network. *Nat. Hazards* **2009**, *50*, 125–143. [[CrossRef](#)]
36. Pereira, S.O.A.F. Modeling spatial patterns of fire occurrence in Mediterranean Europe using Multiple Regression and Random Forest. *For. Ecol. Manag.* **2012**, *275*, 117–129. [[CrossRef](#)]
37. Pourtaghi, Z.S.; Pourghasemi, H.R.; Aretano, R.; Semeraro, T. Investigation of general indicators influencing on forest fire and its susceptibility modeling using different data mining techniques. *Ecol. Indic.* **2016**, *64*, 72–84. [[CrossRef](#)]
38. Moelders, N. Suitability of the Weather Research and Forecasting (WRF) Model to Predict the June 2005 Fire Weather for Interior Alaska. *Weather Forecast* **2008**, *23*, 953–973. [[CrossRef](#)]
39. Kumar, M.; Kosović, B.; Nayak, H.P.; Porter, W.C.; Randerson, J.T.; Banerjee, T. Evaluating the performance of WRF in simulating winds and surface meteorology during a Southern California wildfire event. *Front. Earth Sci.* **2024**, *1*, 1305124. [[CrossRef](#)]
40. Wei, S.; Li, X.; Wang, Z.; Wu, Z.; Luo, S.; Zhou, Y.; Zhong, Y.; Li, Q. Situation and countermeasures of forest fire prevention in Guangdong Province. *Mod. Agric.* **2021**, *10*, 88–90. (In Chinese)
41. NBSC. *China Statistical Yearbook*, 2023; China Statistics Press: Beijing, China, 2023.
42. Zhao, L.; Ge, Y.; Guo, S.; Li, H.; Li, X.; Sun, L.; Chen, J. Forest fire susceptibility mapping based on precipitation-constrained cumulative dryness status information in Southeast China: A novel machine learning modeling approach. *For. Ecol. Manag.* **2024**, *558*, 121771. [[CrossRef](#)]
43. Huete, A.; Didan, K.; Miura, T.; Rodriguez, E.; Gao, X.; Ferreira, L. Overview of the radiometric and biophysical performance of the MODIS vegetation indices. *Remote Sens. Environ.* **2002**, *83*, 195–213. [[CrossRef](#)]
44. Yang, J.; Huang, X. The 30 m annual land cover dataset and its dynamics in China from 1990 to 2019. *Earth Syst. Sci. Data* **2021**, *13*, 3907–3925. [[CrossRef](#)]
45. Estevez, P.A.; Tesmer, M.; Perez, C.A.; Zurada, J.M. Normalized Mutual Information Feature Selection. *IEEE Trans. Neural Netw.* **2009**, *20*, 189–201. [[CrossRef](#)]
46. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [[CrossRef](#)]
47. Ververidis, D.; Kotropoulos, C. Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition. *Signal Process.* **2008**, *88*, 2956–2970. [[CrossRef](#)]
48. Roth, V. The generalized LASSO. *IEEE Trans. Neural Netw.* **2004**, *15*, 16–28. [[CrossRef](#)] [[PubMed](#)]
49. Mustaqim, A.Z.; Adi, S.; Pristyanto, Y.; Astuti, Y. The Effect of Recursive Feature Elimination with Cross-Validation (RFECV) Feature Selection Algorithm toward Classifier Performance on Credit Card Fraud Detection. In Proceedings of the International Conference on Artificial Intelligence and Computer Science Technology, Yogyakarta, Indonesia, 29–30 June 2021; pp. 270–275.
50. Brazdil, P.; Carrier, C.G.; Soares, C.; Vilalta, R. *Metalearning: Applications to Data Mining*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2008.
51. Turner, R.; Eriksson, D.; McCourt, M.; Kiili, J.; Laaksonen, E.; Xu, Z.; Guyon, I. Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*; PMLR: New York, NY, USA, 2021; pp. 3–26.
52. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; De Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* **2015**, *104*, 148–175. [[CrossRef](#)]
53. Feurer, M.; Springenberg, T.; Hutter, F. Initializing Bayesian hyperparameter optimization via meta-learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
54. Reif, M.; Shafait, F.; Dengel, A. Meta-learning for evolutionary parameter optimization of classifiers. *Mach. Learn.* **2012**, *87*, 357–380. [[CrossRef](#)]
55. Gomes, T.A.F.; Prudencio, R.B.C.; Soares, C.; Rossi, A.L.D.; Carvalho, A. Combining meta-learning and search techniques to select parameters for support vector machines. *Neurocomputing* **2012**, *75*, 3–13. [[CrossRef](#)]
56. Wolpert, D.H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259. [[CrossRef](#)]
57. Caruana, R.A.; Niculescu-Mizil, A.; Crew, G.; Ksikes, A. Ensemble selection from libraries of models. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004.
58. He, X.; Zhao, K.; Chu, X. AutoML: A Survey of the State-of-the-Art. *Knowl. Base. Syst.* **2021**, *212*, 106622. [[CrossRef](#)]

59. Pluim, J.P.W.; Maintz, J.B.A.; Viergever, M.A. Image registration by maximization of combined mutual information and gradient information. In Proceedings of the International Conference on Medical Image Computing & Computer-assisted Intervention, Pittsburgh, PA, USA, 11–14 October 2000; pp. 809–814.
60. Thevenaz, P.; Unser, M. Optimization of mutual information for multiresolution image registration. *IEEE Trans. Image Process. A Publ. IEEE Signal Process. Soc.* **2000**, *9*, 2083–2099.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.