International Journal of Intelligent Systems

WILEY

RESEARCH ARTICLE OPEN ACCESS

Meta-Learning Analysis of Deep Neural Network Architectures on Diverse Numeric Datasets via Geometric Complexity Descriptors

Faruk Bulut[1,2] | İknur Dönmez[3]

[1]School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK | [2]Department of Software Engineering, Istanbul Aydin University, Istanbul, Türkiye | [3]Department of Computer Engineering, Istanbul Health and Technology University, Istanbul, Türkiye

Correspondence: Faruk Bulut (faruk.bulut@essex.ac.uk); (farukbulut@aydin.edu.tr)

Received: 16 October 2025 | **Revised:** 28 January 2026 | **Accepted:** 11 February 2026

Academic Editor: Richard Murray

Keywords: accuracy prediction | CNN | complexity measures | dataset geometry | meta-attributes | model selection | transformer

ABSTRACT

Meta-learning techniques aim to predict the most suitable learning algorithm for a given dataset based on its intrinsic structural characteristics. These techniques provide a robust framework for understanding algorithmic behavior across diverse data distributions and attributes. Although these state-of-the-art models (CNNs and transformers) are widely applied in various machine learning tasks, their use on numerical datasets remains underexplored due to the complexity of their internal structures. This study aims not only to predict the performance of two black-box deep learning models on static datasets but also to conduct a behavioral analysis in order to identify which meta-features most strongly influence their outcomes. It seems unclear which specific attributes of a dataset positively or negatively affect the performance of these deep learning models. To bridge this gap, we constructed a meta-dataset consisting of 296 datasets, each characterized by 20 meta-features describing the dataset's statistical, geometric, and structural properties. The analysis identifies which intrinsic dataset properties influence model accuracy, without relying on raw data or hyperparameter tuning. Results show that both models perform best on datasets with high feature discriminability, as captured by meta-features such as maximum feature efficiency, collective feature efficiency, and directional separability. In contrast, performance declines with increasing class boundary complexity and nonlinearity, reflected in features like class separability measures and the linear classifier nonlinearity metric. While CNNs are more sensitive to local geometric complexity, transformers respond more strongly to global statistical measures such as mutual information and entropy, highlighting their distinct inductive biases. The proposed meta-model accurately predicts the performance of both architectures on unseen datasets (0.96 correlation coefficient, 0.019 MAE, and 0.025 RMSE for CNNs; 0.92 correlation coefficient, 0.027 MAE, and 0.036 RMSE for transformers), enabling performance estimation without costly training. These findings emphasize the importance of aligning model architecture with dataset geometry and structure. Additionally, the framework supports more interpretable, efficient, and sustainable deep learning model selection in structured data settings.

1 | Introduction

Selecting the most appropriate classifier for a given dataset is a central challenge in machine learning. Practitioners often need to test multiple classifiers, which can be costly and time-consuming, especially with complex models such as deep neural networks. Meta-learning offers a solution by using knowledge from many datasets to guide model selection and improve

generalization to new tasks [1]. Meta-features, which are quantitative descriptions of a dataset's statistical, geometric, and structural properties, can be used as predictors to estimate the performance of different algorithms [2]. This helps to identify the most suitable algorithm for a dataset and makes model selection more efficient and effective.

Beyond performance prediction, meta-learning provides valuable insights into how different learning algorithms behave under varying dataset characteristics. Analyzing the interaction between dataset properties and model (CNN or transformer) performance contributes to a deeper understanding of both model behavior and learning dynamics. CNNs have demonstrated remarkable success in domains such as image classification. Their ability to learn hierarchical and spatially invariant representations is a major advantage. However, their application to structured numerical and tabular datasets remains relatively underexplored. In such settings, CNN performance strongly depends on the existence of a meaningful feature order and the presence of local feature interactions, conditions that are not always satisfied in tabular data. In parallel, recent advances in meta-learning have increasingly incorporated transformer-based architecture. It is traditionally associated with strong performance in natural language processing and other structured data tasks. Recent studies [3–5] have integrated dataset-level meta-features into transformer models to support model selection and hyperparameter optimization. These approaches demonstrate potential not only for instance-level learning but also for task-level generalization. Nevertheless, transformer-based meta-learning methods continue to face challenges related to interpretability, feature alignment, and computational scalability, underscoring the need for further empirical and theoretical investigation. At the same time, deep learning (DL) models are being actively adapted for classification tasks on numerical and tabular datasets. One-dimensional CNNs (1D-CNNs) process tabular data by treating feature vectors as one-dimensional sequences. Surendro et al. [6] showed that combining 1D-CNNs with principal component analysis (PCA) improves classification performance by reducing redundancy and dimensional complexity. Similarly, in another study [7], accurate multi-radionuclide identification from overlapping gamma-ray spectra remains a challenging problem for conventional analysis methods and existing DL models.

Beyond CNN-based approaches, transformer-based models are increasingly explored as foundation models for tabular learning. By leveraging self-attention mechanisms, transformers can capture complex interfeature dependencies. Hollmann et al. [8] introduced a tabular foundation model that achieves strong predictive performance even on small-scale datasets. They highlight the effectiveness of transfer learning in low-data regimes. Additionally, Jang et al. [9] proposed a meta-transformer framework that integrates meta-learning with transformer architectures for automatic modulation classification, demonstrating scalability and adaptability in structured signal-based tasks.

In general, 1D-CNNs are well suited for tabular datasets with regular feature ordering and strong local interactions, whereas transformers are more effective for high-dimensional, heterogeneous datasets dominated by complex feature interactions. Despite these strengths, both architectures tend to exhibit limitations when applied to small datasets, mixed feature types, or problems involving threshold-based decision boundaries. Given the growing diversity of tabular data (including discrete, sequential, temporal, and spatial forms), there is a need to systematically examine which dataset characteristics favor CNNs or transformers.

In addition to performance limitations, CNNs and transformers are often associated with high computational costs and inefficiencies in feature selection. Feature selection seeks to identify an optimal subset of variables by removing redundant or irrelevant features. However, exhaustive evaluation of all 2^N possible feature subsets for a dataset with N features is computationally infeasible for large-scale problems (NP-hard). To address these challenges, recent studies have increasingly explored metaheuristic optimization algorithms [10]. They have demonstrated strong performance across a range of complex optimization problems due to their global search capabilities and adaptive exploration strategies. Nevertheless, these methods must carefully balance exploration and exploitation to avoid premature convergence or entrapment in local optima. Consequently, metaheuristic and hybrid optimization approaches [11] have been proposed as effective alternatives to classical DL–based methods for tabular data.

Building on these observations, this study hypothesizes that the performance of state-of-the-art black-box DL models, specifically CNNs and transformers, on static numerical datasets is strongly influenced by underlying dataset meta-features. Different dataset characteristics are expected to affect these models in distinct ways, either enhancing or degrading classification performance. By systematically analyzing these relationships, it becomes possible to anticipate model behavior and gain deeper insights into models' internal decision mechanisms.

Accordingly, this study compares the performance of CNNs and transformers on numerical datasets. Here, the objective is to identify the meta-features that most significantly influence classification accuracy. To this end, we construct a comprehensive meta-dataset comprising numerous benchmark datasets with both numerical and categorical attributes, spanning a wide range of real-world applications. This diversity ensures representativeness and enables a robust analysis of how complex black-box architectures operate on structured datasets with well-defined distributions.

2 | Measures for Data Complexity

In meta-learning, basic dataset specifications such as the number of instances, attributes, or class labels are often insufficient for extracting meaningful meta-features. Even when including simple statistics such as min, max, median, variance, or standard deviation, these metrics mainly reflect dataset density or sparsity and provide only a general overview of the feature space. To improve predictive performance, more sophisticated meta-features are needed to capture the dataset structure.

The reviews by Refs. [2, 12, 13] highlight the importance of systematizing meta-features due to inconsistencies in their description and computation. It introduces a taxonomy for meta-features, enhancing clarity and utility, and recommends further research on regression, clustering, and interpretability. Ho and Basu [14] emphasize geometric characteristics, such as class distributions, that influence classifier performance. Classes may be well-separated, interleaved, linearly separable, or overlapping, and fourteen geometric measures from Ref. [15] are used in this study to capture data complexity and predict classifier performance.

Some initial statistical meta-features such as "imbalance ratio," "instance-to-feature ratio," and "mean feature entropy" were tested but later removed as they proved computationally intensive and ineffective for DL models. The most effective features used in our experiments are described below.

2.1 | Feature-Based Metrics for Class Overlaps (F1, F1v, F2, F3, and F4)

F1 (Fisher's discriminant ratio) is a global measure used to assess the maximum discriminative power of each feature in a dataset, determined by

$$F1 = \max_{i=1}^{l} f_i \text{ and } f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 - \sigma_2^2}, \tag{1}$$

where l represents the number of features. f_i is the discriminant ratio of each attribute. μ_1 and μ_2 are the means, and σ_1^2 and σ_2^2 are the variances of the two classes for a given attribute. These parameters describe the class distributions and their separation. The value of f is computed for each feature dimension, and the maximum f value across all dimensions is selected as F1. A high F1 value indicates that at least one attribute allows effective class separation with partitions parallel to an axis of the feature space [16].F1v (directional-vector max. F1) is derived from F1 and serves as its complement, incorporating an oriented vector capable of separating samples from different classes. It calculates the binary-class Fisher's criterion as

$$R(d) = \frac{\left[\vec{d}^2(\vec{\mu}_1 - \vec{\mu}_2)\right]^2}{\vec{d}^T \overline{\sum} \vec{d}} = \frac{\vec{d}^T B \vec{d}}{\vec{d}^T \overline{\sum} \vec{d}}, \tag{2}$$

where $\vec{\mu}_i$ and $\vec{d}$ are the mean vector and the directional vector, respectively. $\sum_i$ is the scatter matrix of patterns for class c_i. $B = (\vec{\mu}_1 - \vec{\mu}_2) * (\vec{\mu}_1 - \vec{\mu}_2)^T$ is between class scatter matrix. $\vec{d}$ is computed as $\vec{d} = \overline{\sum^{-1}} \Delta$ where $\Delta = \mu_1 - \mu_2$ and $\overline{\sum^{-1}}$ is computed as the pseudo inverse of $\overline{\sum}$. A higher value indicates the presence of a vector capable of separating samples from different classes.

Volume of the overlapping region (F2) quantifies the overlap between classes by measuring the overlap interval of each feature and multiplying these ratios across all dimensions. Higher F2 implies stronger class overlap.

Maximum feature efficiency (F3) measures how well a single feature separates classes by computing the proportion of samples that do not overlap along that feature. It is limited to axis-aligned decision boundaries.

Collective feature efficiency (F4) evaluates the combined discriminative power of all features. Features are applied iteratively to separate samples, and the final score represents the proportion of correctly separated instances. Similar to F3, F4 relies on hyperplanes parallel to feature axes, but unlike F3, it evaluates the joint effectiveness of all features rather than individual ones.

2.2 | Class Separability Measures (N1, N2, and N3)

This section examines the shape of class boundaries. Firstly, N1 (the fraction of points located on the class boundary) quantifies the extent of the class boundary within a dataset. To compute N1, a minimum spanning tree (MST) is constructed using algorithms such as Kruskal's or Prim's, connecting each point to its nearest neighbor. N1 is then defined as the ratio of edges linking points from different classes to the total number of points.

Figure 1 illustrates the N1 calculation on a binary-class dataset with 16 nodes. Since darker edges represent the connections between different classes, N1 = 9/16. Low N1 values indicate that the class boundary is easier to define, whereas high N1 values suggest that many points lie near the boundary, increasing classification complexity [17].

N2 (Ratio of average intraclass to interclass NN distance) evaluates the relative spread of samples within classes compared to the distance to nearest neighbors from other classes. For each instance x_i, the distance to its nearest neighbor in the same class (intraDist(ex_i)) and the distances to its nearest neighbor from a different classes (intraDist(ex_i)) are computed. N2 is defined as

$$N2 = \frac{\sum_{i=0}^{N_e} \text{intraDist}(ex_i)}{\sum_{i=0}^{N_e} \text{interDist}(ex_i)}, \tag{3}$$

where N_e is the total number of examples. Lower N2 values indicate that same-class samples are tightly clustered, while higher values suggest a more dispersed distribution across the feature space.

N3 is the leave-one-out error rate of the 1NN classifier, measuring how close samples from different classes are to each other. A low N3 value indicates well-separated classes, while a high value suggests overlapping class boundaries.

2.3 | Linear Classifier Errors (L1, L2, and L3)

L1 (minimized sum of error distance for linear classifier) is a measure for binary classification that evaluates whether a dataset is linearly separable. An L1 value of zero indicates perfect separability. It is computed by summing the differences between linear classifier predictions and actual labels while minimizing the error function:

$$\text{minimize } a^t t \longrightarrow \text{subject to } Z^t w + t \geq b, \quad t \geq 0, \tag{4}$$

where a and b are the constant vectors (commonly set to 1), w is the weight vector, t is the error vector, and Z is a matrix with columns defined by input vectors. Class labels are assigned as

$$\begin{aligned} z &= +x \text{ if } c = c_1, \\ z &= *x \text{ if } c = c_2. \end{aligned} \tag{5}$$

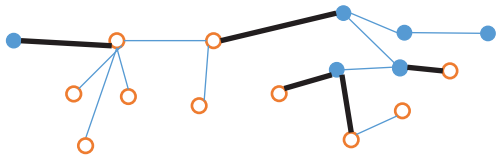


FIGURE 1 | An illustration of N1 calculation.

L1 can be influenced by outliers and noisy data.

L2 (the training error for a linear classifier) is another measure specifically designed for binary-classification tasks. To calculate L2, a classical linear classifier is initially trained. The classifier's training error rate is then calculated, indicating the degree of linear separability within the training data.

L3 (linear classifier nonlinearity) is designed for two-class datasets and measures the nonlinearity of a linear classifier as seen in Figure 2. A new test set is created by linearly interpolating between randomly selected pairs of instances from the same class using random coefficients. The test error of an SVM trained on the original data is then evaluated. This measure is especially sensitive to class overlap within the convex hull and to the smoothness of the classifier's decision boundary.

2.4 | Geometric, Topological, and Density Measures for Manifolds (N4, T1, and T2)

This measure targets two-class datasets and quantifies the nonlinearity of a linear classifier. It generates a new test set by linearly interpolating between randomly chosen pairs of instances from the same class using random coefficients and then evaluates the test error of a support vector machine (SVM) [18] trained on the original data. This approach is particularly sensitive to class overlap within the convex hull and to the smoothness of the classifier's decision boundary.

N4 (the nonlinearity of 1NN), like the nonlinearity measure of a linear classifier (L3), involves creating a test set through interpolation. It then calculates the error rate of a 1NN classifier on this test set.

T1 (The fraction of maximum covering spheres) describes the shapes of class manifolds by using adherence subsets. An adherence subset is a sphere centered on a sample that expands as much as possible without including samples from other classes. Each subset contains only samples from the same class and cannot grow further without intersecting samples from another class. T1 focuses on the largest such spheres, excluding those contained within others, and computes their number normalized by the total number of points.

As a general indicator, T2 (the average number of points per dimension) estimates the dataset's density by calculating the ratio of the number of samples to the number of attributes. When considering the dataset as a matrix, T2 represents the ratio of rows to columns.

2.5 | Statistical Characterization of Datasets ($H(D)$, feature correlation [FC], MaxMI, and MMI)

In the literature, there are some other fundamental measures, such as entropy and average FC, which quantify the information content of a dataset and the dependency among its features, respectively [19].

Entropy $H(D)$ quantifies the amount of "information disorder" in a dataset. A dataset with all samples belonging to the same class has entropy = 0 (perfectly pure). A dataset with an equal distribution of classes has maximum entropy (highest uncertainty). The entropy of a dataset D is calculated as below, where C is the number of distinct classes and p_i is the proportion of samples in class:

$$H(D) = -\sum_{i=1}^{C} p_i \log_2(p_i). \tag{6}$$

Average FC measures the overall linear dependency between the features of a dataset. It gives an idea of how redundant or independent the features are. If many features are highly correlated, the dataset may contain redundant information. If correlations are low, features are more independent, which often helps classifiers learn better. Let us suppose a dataset has d numerical features, and then, FC is then computed as

$$\mathrm{FC} = \frac{2}{d(d-1)} \sum_{j=1}^{d-1} \sum_{k=j+1}^{d} \left| \frac{\mathrm{Cov}(X_j, X_k)}{\sigma_{X_j} \times \sigma_{X_k}} \right|. \tag{7}$$

In the vertical bars, there is the Pearson absolute correlation coefficient between two features where $\mathrm{Cov}(X_j, X_k)$ is the covariance between the features X_j and X_k. Also, σ_{X_j} and σ_{X_k} are the standard deviations of the features.

If FC is close to 0, it indicates that the features are largely uncorrelated, suggesting that the dataset contains mostly independent information. If FC is close to 1, it indicates that the features are highly correlated, implying the presence of redundancy in the dataset.

Mutual information (MI) quantifies the dependency between a feature and the class label. For a dataset with d features $X_1, X_2, \ldots, X_d$ and class labels C, MI is computed as

$$\begin{aligned} \mathrm{MI}(X_j, C) &= \sum_{x \in X_j} \sum_{y \in Y} p(x, c) \log_2 \left(\frac{p(c, y)}{p(c)p(y)} \right) \\ &= \max_{j=1,..,d} \mathrm{MI}(X_j, C), \text{ and } \mathrm{MMI} = \frac{1}{d} \sum_{j=1}^{d} \mathrm{MI}(X_j, C). \end{aligned} \tag{8}$$

MaxMI is the maximum MI of a feature with the class, while mean MI (MMI) is the average across all features. MaxMI highlights the strongest single predictor, MMI shows overall informativeness, and their gap indicates whether predictive power is concentrated in a few features or more evenly distributed.

2.6 | Time Complexity Analysis and Categorization

Table 1 presents 14 geometrical and three statistical meta-features along with their corresponding time complexities in big-O notation. In this context, n denotes the number of input samples, n_t represents the number of test samples (used only for meta-features requiring an additional test set), c is the number of class labels, and a is the number of attributes. The time complexity column provides the computational complexity for each meta-attribute, which is essential for assessing the computational resources needed, particularly for large-scale datasets. The notation O (SMO) refers to training a linear-kernel SVM using the sequential minimal optimization (SMO) algorithm [20].

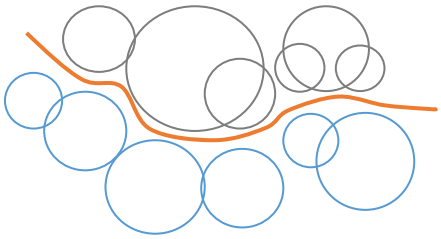


FIGURE 2 | Adherence subsets retained near the boundary for two classes.

3 | Methods, Data, and Feature Extraction

The flowchart in Figure 3 illustrates the workflow of the proposed meta-learning system. It starts with raw datasets, which are preprocessed before analysis. A correlation coefficient matrix captures feature dependencies, and meta-feature vectors describes the datasets' structural, geometrical, and statistical properties. The classification performance results (accuracy) of 1D-CNN and transformer models were evaluated using k-fold cross-validation and put to the *Meta*-dataset as class labels. Finally, the regression models trained on this *meta*-dataset are used to generalize knowledge, producing insights labeled "Wisdom" in the diagram as in the DIKW pyramid. This stage highlights the ability of the system not only to predict model performance but also to provide interpretable guidance on the suitability of CNNs or transformers for given datasets.

3.1 | 1D-CNN Architecture

CNNs are a class of DL models particularly effective in learning local patterns through convolutional operations and have been extensively applied in fields such as computer vision, signal processing, and time series classification [21]. While traditional CNNs operate on two-dimensional image data, 1D-CNNs are designed to process 1D sequences, making them suitable for numerical datasets, temporal data, or feature vectors where spatial locality and ordering are relevant. In this study, we utilize a 1D-CNN architecture adapted for structured numerical datasets. The architecture is composed of convolutional layers for feature extraction and fully connected layers (multilayer perceptrons or MLPs) [22] for final classification. Key architectural components of the CNN used in this study can be listed as follows:

- Convolutional Layers: These layers apply a set of learnable filters (kernels) to local subsequences of the input data. Each filter detects specific local patterns across the sequence. In our base model, three convolutional layers are used.
- Subsampling/Pooling Layers: Pooling operations (such as max-pooling or average-pooling) reduce the dimensionality of feature maps, improve computational efficiency, and introduce spatial invariance. In this study, the downsampling factor after each CNN layer is 4.
- Activation Functions: Nonlinear activation functions (e.g., ReLU, hyperbolic tangent [tanh]) [23] are applied to introduce nonlinearity into the model, enabling the network to learn complex patterns.
- Fully Connected Layers (MLPs): After the convolutional layers, the high-level features are flattened and passed to one or more fully connected layers for final classification. Two fully connected layers are used in our base model.
- Filter Size: The kernel width of each CNN layer is 41 in this study.

The performance and generalization ability of a 1D-CNN are highly dependent on its hyperparameter configuration. To facilitate performance prediction via meta-learning, we record the architectural hyperparameters of each CNN model, as illustrated in Figure 4. These parameters not only shape the model's capacity and learning dynamics but also serve as input variables for our meta-learning framework. By associating different configurations with their corresponding classification performance across datasets, we aim to uncover systematic relationships between model architecture and dataset properties.

The hyperparameters used in this study include filter sizes {32, 64, 128}; kernel sizes {3 | 5 | 7}; strides {1 | 0}; activation functions ReLU for hidden layers and softmax or sigmoid for the output layer; optimization with adam (learning-rate, LR = 0.001) or sgd (LR = 0.01); dropout rates [0.2, 0.5]; batch normalization applied after convolutional layers; batch sizes {32 | 64}; and training epochs [20, 100].

3.2 | Transformer Architecture

The classifier architecture was based on the transformer encoder paradigm, adapted for tabular data. Specifically, the model comprises three key components: (1) an input embedding layer that linearly projects the raw feature vector into a fixed-dimensional representation space ($d\_model = 64$), (2) a transformer encoder block consisting of a single n, TransformerEncoderLayer with four self-attention heads ($nhead = 4$), and (3) a fully connected output layer mapping the encoded representation to two logits corresponding to the binary classes. The transformer encoder enables the model to capture global dependencies across input features via multihead self-attention, which is particularly beneficial for datasets with complex, nonlocal feature interactions [24].

Model training was performed using the Adam optimizer [25] with a fixed LR of 0.001 and cross-entropy loss as the objective function. Each model was trained for 10 epochs on its respective dataset without early stopping or regularization in order to maintain consistency across experiments. At inference, the trained models were evaluated on the test set, and accuracy metrics were recorded for each dataset. Additionally, standard classification reports, including precision, recall, and F1 score, were computed to provide a more granular assessment of model performance.

This experimental framework allowed for a systematic investigation of the transformer's behavior across a wide variety of tabular data distributions and feature characteristics, offering insights into the model's generalization capabilities and sensitivity to dataset-level meta-features. The model used in this study is as illustrated in Figure 5.

The transformer classifier parameters used in this study include two to six encoder layers, embedding dimensions (d_model) of 128, 256, or 512, and four or eight attention heads. The feed-forward size (dff) ranges from 512 to 2048, with dropout rates between 0.1 and 0.3. The optimizer is *adam* or *adamw*, with LRs

TABLE 1 | List of time complexities for each of the meta-features.

	Definition	Time complexity	Category	Subtype	Interpretation
F1	Fisher's discrimination ratio	$O(n.a)$	Feature based	Separability	Measures class separability based on feature means
F1v	Directional-vector max Fisher's discriminant ratio	$O(n.a + a^3)$	Feature based	Directional separability	Captures directional class separability
F2	Volume for overlap region	$O(n.a)$	Feature based	Overlap measurement	Estimates region where classes overlap
F3	Max value of feature efficiency	$O(n.a.c)$	Feature based	Feature relevance	Evaluates individual feature efficiency
F4	Collective feature efficiency	$O(n.a^2.c)$	Feature based	Global relevance	Evaluates global feature relevance
N1	Fraction of points over class boundary (MST)	$O(n^2.a)$	Neighborhood based	Graph separability	Graph-based measure of boundary complexity
N2	Ratio of avg intra-/interclass NN distance	$O(n^2.a)$	Neighborhood based	Distance ratio	Separability via distance ratio
N3	LOO CV error of 1NN classifier	$O(n^2.a)$	Neighborhood based	Validation error	Nearest-neighbor generalization error
L1	Min sum of error distance for linear classifier	$O(SMO)$	Linearity based	Margin error	Measures margin error of linear classifier
L2	Linear classifier's training error rate	$O(SMO)$	Linearity based	Misclassification	Training error of linear classifier
L3	Linear classifier nonlinearity	$O(SMO + n_t.a.c)$	Linearity based	Nonlinear boundary	Measures degree of nonlinearity in linear decision boundary
N4	Nonlinearity of 1NN classifier	$O(n_t.a.c + n.n_t.a)$	Neighborhood based	Boundary complexity	Nonlinearity via nearest-neighbor rules
T1	Fraction of points with retained adherence subsets	$O(n^2.a)$	Topological	Class manifold adherence	Measures manifold complexity
T2	Dataset density estimation (sample-to-attribute ratio)	$O(1)$	Statistical	Density metric	Estimates dataset sparsity/density
$H(D)$	Entropy: Information disorder	$O(c)$	Statistical	Uncertainty	Captures impurity/randomness of labels
FC	Average feature correlation	$O(c^2.n)$	Statistical	Redundancy analysis	Measures redundancy among features
MI	Mutual information (MinMI, MMI)	$O(n.c)$	Statistical	Dependency measure	Captures dependency between features and labels

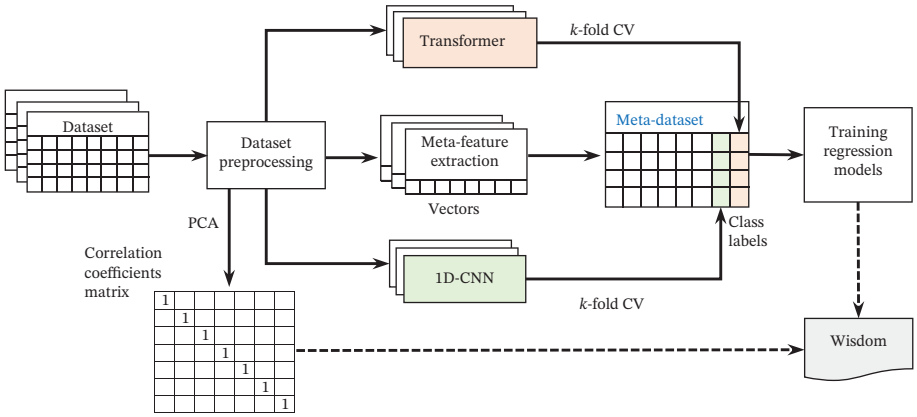


FIGURE 3 | The flowchart of the proposed model.

of 1e − 4 to 1e − 5 for fine-tuning and 1e − 3 for training from scratch. Activation functions include ReLU or GELU, and the loss function is *categorical_crossentropy* or *binary_crossentropy*. The model is trained with batch sizes of 32 or 64 over 10–50 epochs, using metrics such as accuracy, F1 score, and precision.

As can be seen above, the architectural details and hyperparameters of both the CNN and transformer models are explicitly provided. The primary aim of this study is to construct equivalent and balanced parameter settings for both classifiers and to compare their performances under fair and identical experimental conditions. Accordingly, the models were trained using comparable batch sizes and training epochs to ensure a controlled and unbiased evaluation. This design choice allows the observed performance differences to be attributed to the intrinsic characteristics of the classifiers and the dataset meta-features rather than to disparities in training configurations [26].

3.3 | Dataset Selection and Preprocessing

A total of 296 binary-class, multivariate, non–time-series datasets were selected, with the number of attributes ranging from 2 to 1177 and instances between 36 and 100,000. Both numerical and categorical features were included. The datasets were drawn from diverse real-world domains, including computer science, healthcare, finance, economics, education, management, entertainment, transportation, cryptocurrency, linguistics, literature, and life sciences, ensuring broad representativeness. To enable DL models to be trained over a wider spectrum and to produce more general and comprehensive hypotheses, datasets were deliberately collected from different environments and diverse domains, thereby forming a broad-spectrum data pool. Our aim was also to ensure that DL models could be trained in a way that makes them adaptable and robust across various scenarios. This preprocessing ensured that the resulting meta-features were reliable and meaningful for meta-learning analyses. Among the datasets listed, approximately 150 are focused on health or medical topics (including *breast cancer*, *diabetes*, *heart disease*, *hepatitis*, *thyroid*, *liver disorders*, and related clinical datasets). Around 30 datasets pertain to finance or business-oriented domains (such as bank marketing, credit, personal loans, and customer churn). Another 25 datasets are related to computer science, signal processing, or sensor-based domains (including *sonar*, *ionosphere*, *waveform*, *segment*, and *movement* datasets). The remaining datasets are distributed across manufacturing, government, physical science, social, or education-related areas, with only a few in purely educational datasets.

Since categorical variables have no natural order, one-hot encoding was used to represent each category without creating a false ranking. Missing values were handled using simple statistical imputation: Numerical features were filled with their mean values because the amount of missing data was low, which helps keep the overall data structure while remaining efficient. Missing categorical values were replaced with the most common category. These preprocessing steps make the DL pipeline clearer, more reliable, and easier to reproduce, in line with standard practices in the literature [27].

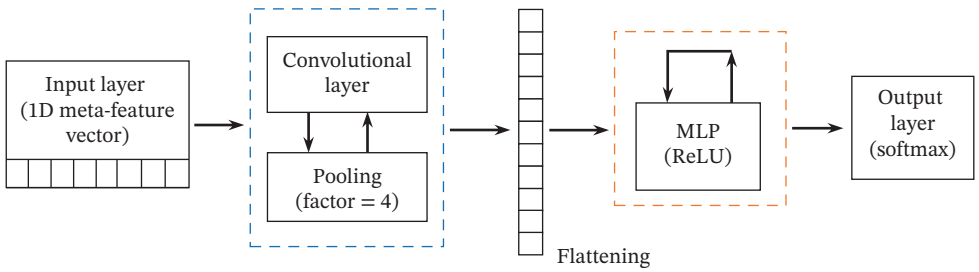


FIGURE 4 | Architecture of the 1D-CNN used in our study.

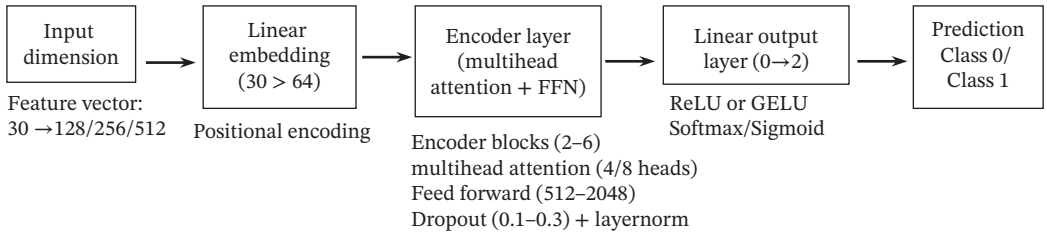


FIGURE 5 | Architecture of the transformer used in our study.

Among the 296 datasets, 41 originated from Kaggle [28] and the remainder primarily from UCI [29]. Of these, 98 were originally binary-class datasets, while 157 multiclass datasets were transformed into binary-class using a one-vs-rest approach. Only balanced binary-class transformations were retained, avoiding highly imbalanced datasets that could lead to misleading accuracy, such as cases where a trivial classifier like ZeroRule would achieve near-perfect accuracy on a majority-class dataset. Datasets originally split into training and testing partitions were merged, as *k*-fold cross-validation was applied. For consistency, an 80/20 stratified train–test split was used across all datasets. Missing values were imputed, categorical features numerically encoded, and all features normalized via Z-score standardization. Datasets in.dat, .arff, or. csv formats were converted to KEEL [30] format for use in MATLAB R2025a, DCoL [31] (running in Linux Ubuntu OS), Weka [32], and Python platforms using PyTorch and TensorFlow libraries. All experimental studies were conducted on a computer with an AMD EPYC7302P 16-Core processor running at 2.99 GHz and 8 GB RAM. All models were trained and evaluated on the same hardware environment to ensure a fair and consistent comparison. Moreover, to guarantee experimental fairness, identical training configurations were used across all models, including the same number of training epochs and batch sizes, unless explicitly stated otherwise.

3.4 | Meta-Feature Extraction

Structural, statistical, information-theoretic, and model-based meta-features were analyzed to capture key differences across datasets. Structural aspects included the number of instances, features, and the ratio of categorical to numerical attributes. Entropy values were considered to reflect information-theoretic diversity, while statistical variation was measured through differences in feature variance. Model-based differences were also included, since some datasets were well handled by simple models, while others were not. Emphasis was placed on ensuring these contrasting characteristics were well represented for broad coverage.

Ten selected datasets are presented in Table 2 along with their extracted meta-features used in this study. Each row corresponds to a specific dataset, and each column represents a particular characteristic of that dataset. Only a subset of the full collection of datasets is shown for illustration. The columns include basic external specifications such as the dataset source (S: U = UCI, K = Kaggle), the number of features (nF), and the number of samples (nS). Additional columns capture statistical and geometrical properties, including $H(D)$, MMI, MaxMI, FC, and a set of 14 meta-features that quantify class separability, linearity, dataset geometry, and density characteristics. All datasets are publicly available at the URLs provided in the Data Availability Statement section. For meta-learning, these features were treated as input attributes, while the classification accuracy of CNN or transformer models on the corresponding dataset was used as the target variable. To ensure comparability across datasets, all meta-features were normalized between 0 and 1. This normalization enables the direct interpretation of regression coefficients, facilitates the understanding of their relationship with accuracy, and provides insight into how various dataset characteristics influence the performance of CNN and transformer classifiers.

4 | Experimental Studies and Analysis of Results

The experimental study followed a structured procedure. Numerical datasets were first selected and preprocessed, and relevant meta-features were systematically extracted. CNN and transformer classification accuracies were then computed for each dataset. Using this information, a linear regression model [33] was developed to predict expected performance on unseen datasets.

Beyond prediction, the study examines how meta-features impact model performance and seeks to understand the factors that drive the accuracy of CNN and transformer models. Comparative analysis highlights conditions where one model outperforms the other, providing guidance on model selection based on dataset characteristics.

4.1 | Accuracy Forecasting for Classifiers

Table 3 represents the predicted accuracies of CNN and transformer models, along with their absolute errors. Due to space limitations, only 10 out of the 296 datasets are presented. The results show that performance differs across datasets. For example, on the *cervical_cancer* dataset, both CNN and transformer achieve very high accuracy with minimal error, indicating that the models can effectively capture the underlying data structure. In contrast, on the *smoke_detection_iot* dataset, CNN produces a large prediction error (0.0501), while the transformer maintains strong performance (0.0035), highlighting its robustness in handling these types of data. Conversely, in the *liver-disorder* dataset, CNN struggles with a higher error (0.0557), whereas the transformer achieves much lower error (0.0089). There are also cases where both models underperform, such as in the *bpa* dataset, reflecting challenges posed by the inherent complexity or noise in the data. These differences emphasize that no single model consistently dominates across all datasets, aligning with the no free lunch theorem.

In our study, we first ran the basic CNN and transformer classification algorithms mentioned in the Methods, Data, and Feature Extraction section to find the actual classification accuracy of each dataset. Next, using these accuracy values along

TABLE 2 | Meta-features of datasets.

Datasets	*S*	*nF*	*nS*	*H* (*D*)	MMI	MaxMI	FC	F1	F1v	F2	F3	F4	L1	L2	L3	N1	N2	N3	N4	T1	T2
Mushroom	U	112	8124	1.000	0.040	0.370	0.100	3.31	1590	0.000	0.266	0.99	0.355	0.001	0.003	0.003	0.39	0.000	0.003	1.00	72.5
ShillBidding	K	9	6321	0.490	0.092	0.411	0.211	9.97	46.23	0.788	0.433	0.48	0.811	0.026	0.026	0.020	0.13	0.008	0.048	0.91	702.3
adl	U	14	48,842	0.794	0.022	0.090	0.059	0.35	5.36	0.233	0.030	0.03	1.064	0.166	0.278	0.290	0.45	0.201	0.270	0.99	3488.7
bpa	U	6	345	0.982	0.036	0.080	0.265	0.06	0.15	0.073	0.032	0.11	0.841	0.420	0.500	0.574	0.91	0.374	0.343	1.00	57.5
BankAdditional	K	20	41,188	0.508	0.026	0.088	0.221	0.58	5.40	0.140	0.051	0.05	0.283	0.103	0.400	0.186	0.48	0.127	0.306	0.99	2059.4
cervical_cancer	K	35	858	0.706	0.016	0.418	0.165	4.96	25.2	0.000	0.141	0.22	0.090	0.037	0.090	0.100	0.34	0.064	0.164	0.93	24.5
loan_data	K	13	45,000	1.000	0.007	0.038	0.192	0.40	39.4	0.000	0.508	0.51	1.790	0.110	0.140	0.191	0.22	0.126	0.199	1.00	3461.5
diabetes_prediction	K	8	100,000	2.140	0.030	0.156	0.134	1.13	3.40	0.024	0.418	0.62	1.715	0.040	0.208	0.076	0.15	0.050	0.148	0.99	12,500.0
credit-g	U	59	1000	0.880	0.010	0.060	0.060	0.31	1.86	0.662	0.014	0.02	0.744	0.231	0.348	0.447	0.89	0.283	0.125	1.00	16.9
smoke_detection_iot	K	15	62,630	0.863	0.205	0.644	0.343	2.82	−0.32	0.008	0.615	0.97	3.295	0.040	0.032	0.000	0.01	0.001	0.047	0.92	4175.3

TABLE 3 | CNN and transformer accuracy results according to meta-features.

Datasets	CNN			Transformer		
	Actual	Prediction	Absolute error	Actual	Prediction	Absolute error
mushroom	1.0000	0.9659	0.0341	0.9975	0.9623	0.0353
ShillBidding	0.9779	0.9577	0.0201	0.9731	0.9515	0.0216
adl	0.8446	0.8269	0.0177	0.8217	0.8426	0.0209
bpa	0.6377	0.6400	0.0023	0.7246	0.7042	0.0205
BankAdditional	0.9099	0.8842	0.0258	0.9014	0.8891	0.0123
cervical_cancer	0.9328	0.9317	0.0011	0.9254	0.9152	0.0101
loan_data	0.9016	0.8724	0.0292	0.8769	0.8867	0.0098
diabetes_prediction	0.9462	0.9142	0.0320	0.9290	0.9363	0.0074
credit-g	0.7650	0.7601	0.0049	0.7650	0.7613	0.0037
smoke_detection_iot	0.9998	0.9498	0.0501	0.9586	0.9551	0.0035

with extracted meta-features, we generated a model to predict the accuracy. Once the *meta*-dataset was constructed, it served as a basis for training various classification or regression algorithms. These trained models were subsequently employed to predict the performance of CNN and transformer classifiers on the existing datasets using solely the extracted meta-features. The experimental findings demonstrated encouraging results, as seen in Table 3.

Figure 6 shows the accuracy performance of the CNN and transformer classifiers across the datasets. For comparison, datasets were sorted in ascending order by their actual accuracy. The continuous red line marks the predicted accuracy rates. Discrete blue dots indicate the actual calculated accuracy for each dataset. The x axis shows the dataset indices after sorting by actual accuracy. This order does not imply any relationship between neighboring datasets. Visual deviations (ripples) between predicted and actual values represent the prediction errors. The strong alignment between predicted and actual values highlights the effectiveness of the regression model.

The regression model developed through this meta-learning process is particularly valuable, but not enough; we want to delve into the underlying factors related to this classification performance. By analyzing the contributions of the 14 meta-features in the regression model, it will be possible to identify specific features of the datasets that lead to better or worse classifier performance.

4.2 | CNN Accuracy Estimation With Meta-Features

In this study, we aimed to understand the relationship between dataset meta-parameters and prediction accuracy for CNNs. This approach involves evaluating the statistical and predictive relevance of each meta-parameter for CNN accuracy, specifically using various feature importance and correlation analysis techniques. The goal is to quantitatively evaluate the extent to which each data meta-parameter impacts CNN model accuracy using both linear and nonlinear statistical methods. The feature importance and correlation analysis techniques used can be defined for the analysis of CNN and transformer as follows.

Ensemble-based feature importance (random forest, RF): CNN accuracy is estimated from the meta-parameter space using a RF regressor [34]. The trained ensemble model is used to determine feature importance, which shows how often and successfully each feature is applied in decision trees to lower prediction error.

Nonlinear Dependence (MI): The MI between each feature and the target variable is estimated using the mutual-info-regression function. By capturing nonlinear dependencies, this approach is appropriate for detecting intricate, nonmonotonic interactions that linear models might overlook.

Linear Correlation (Pearson Coefficient): X. corrwith (y) is used to get the Pearson correlation coefficient [35] for each feature and the target variable. The strength and direction of linear relationships are measured by this technique, which goes by the designation r. r values near $+1.0$ indicate a strong positive relation, near -1.0 a strong negative, and near 0.0 little or none.

Univariate Linear Regression (F-Regression) [36]: Using the F-regression method, each meta-parameter (feature) in the X matrix is regressed separately against the target variable y (classifier accuracy). The resulting F-statistic and p value quantify the strength and significance of the linear relationship between each meta-parameter and classifier performance.

We used four complementary statistical methodologies to evaluate the relationship between the dataset's meta-parameters and the predicted accuracy of the model, as indicated in Table 4. These comprise a model-based strategy (RF feature importance) as well as model-independent statistical techniques (F-regression, Pearson correlation, MI, and RF importance) [37]. By emphasizing the characteristics that most affect accuracy, these findings help determine or interpret the meta-parameters that work best for the classifiers.

The analysis presented in Table 5 highlights the relationship between various meta-features and classification accuracy using multiple importance metrics. Among the evaluated features, L2, N1, and N3 exhibit consistently strong negative correlations with accuracy (e.g., L2: $r = -0.75$, N1: $r = -0.71$), suggesting that higher values of these meta-features are associated with decreased CNN classifier performance. These meta-features also score high in RF importance and F score, indicating their

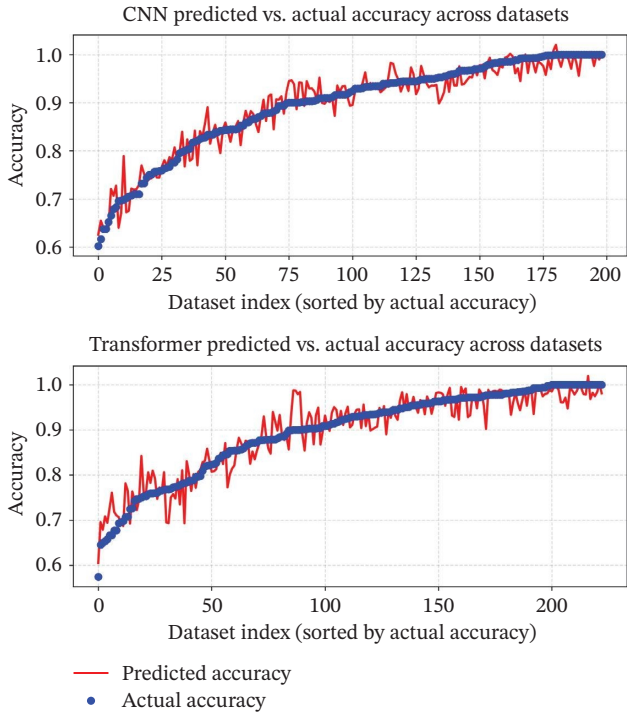


FIGURE 6 | Diagrams for actual and predicted accuracy values of CNN and transformer.

substantial influence on model behavior. Conversely, features such as F3 and F4 demonstrate positive correlations with accuracy (e.g., F3: $r = 0.44$), implying that they may contribute positively to classification performance. Moreover, several features with statistically significant F scores (e.g., L1, L3, H (D)) exhibit moderate to strong negative correlations, reinforcing their relevance for characterizing problem difficulty. These findings suggest that instance hardness measures (e.g., N1–N4, L2) and data complexity descriptors (e.g., H (D), L1) are critical in predicting classification success and should be considered in meta-learning and dataset characterization frameworks.

We used a linear regression model to study how meta-features affect CNN accuracy, using dataset-level information instead of raw data. The model showed good performance, with low RMSE and mean absolute error (MAE) and a high correlation coefficient. To improve the reliability of the results, we first checked the p values and included only the meta-features that were statistically significant.

The first baseline model applies the ZeroRule, where no meta-features are used. As expected, this yields a very weak correlation coefficient (−0.1637), with relatively a high MAE (0.0764) and RMSE (0.0958). This baseline highlights the necessity of including dataset-level meta-features to obtain meaningful CNN accuracy predictions.

In the second formula, CNN accuracy is influenced by a wider set of features, combining statistical (MMI, F2–F4), linear separability (L1, L3), neighborhood measures (N2–N4), and structural adherence (T1). Among these, N4 and F3 have positive contributions, indicating that certain feature-specific descriptors and structured neighborhoods enhance CNN performance. However, strong negative effects are observed for N3, L3, MMI, and L1, reflecting the detrimental role of local irregularities, higher-order linear errors, and weak MI. The intercept (1.0214) provides a solid baseline, but the overall correlation (0.8593) and error rates (MAE 0.0344, RMSE 0.0501) show this model is less effective compared to more compact alternatives.

The third regression model achieves much stronger predictive power, with a correlation coefficient of 0.9644 and very low error values (MAE 0.0201, RMSE 0.0253). Here, L2 and N1 stand out as strong negative contributors, highlighting that poor linear separability and class overlap are the main obstacles for CNN accuracy. In contrast, N2 and N3 appear as positive contributors, suggesting that structured neighborhood information can sometimes aid performance. L1 provides a small positive effect, while entropy (H) negatively affects accuracy, consistent with the idea that disorder in class distribution reduces learnability. The intercept (0.9965) indicates a high baseline accuracy in the absence of these challenges.

TABLE 4 | Significant meta-features influencing CNN accuracy ($p < 0.05$).

Feature abbrv.	Random forest importance	MI (nonlinear relationship)	Correlation of accuracy (r)	Univariate feature selection	
				F score	p value
F1v	0.0252	0.1888	0.1720	8.05	0.0049
F2	0.0041	0.1830	−0.2161	12.93	0.0004
F3	0.0107	0.2887	0.4448	65.11	0.0000
F4	0.0107	0.2860	0.3607	39.49	0.0000
L1	0.0160	0.2620	−0.2027	11.31	0.0009
L2	0.4364	0.7738	−0.7525	344.60	0.0000
L3	0.0084	0.1710	−0.2460	17.00	0.0001
H (D)	0.0211	0.4382	−0.3339	33.12	0.0000
N1	0.1799	0.5961	−0.7137	274.08	0.0000
N2	0.0142	0.2458	−0.5357	106.27	0.0000
N3	0.0502	0.5680	−0.6626	206.63	0.0000
N4	0.0113	0.1929	−0.4116	53.85	0.0000
T1	0.0092	0.1006	−0.1523	6.27	0.0129

TABLE 5 | Equations for CNN accuracy with different meta-features.

Input parameters	Formula for CNN accuracy with selected parameters	Corr. coef.	MAE	RMSE
All parameters	ZeroRule (*The worst performance*)	−0.1637	0.0764	0.0958
All parameters	$\mathrm{Acc_{CNN}} = -0.0885\mathrm{MMI} + 0.0147\mathrm{F2} + 0.0014\mathrm{F3} + 0.0039\mathrm{F4} + -0.0172\mathrm{L1} + -0.1142\mathrm{L3} + -0.0035\mathrm{N2} + -0.6625\mathrm{N3} + 0.0476\mathrm{N4} + -0.0164\mathrm{T1} + 1.0214$	0.8593	0.0344	0.0501
Only parameters with $p < 0.05$	$\mathrm{Acc_{CNN}} = -0.0134\mathrm{H} + -0.0169\mathrm{F3} + 0.0138\mathrm{L1} + -0.5307\mathrm{L2} + -0.3992\mathrm{N1} + 0.0399\mathrm{N2} + 0.1674\mathrm{N3} + 0.9965$	0.9644	0.0201	0.0253
F2, L2, L3, N1, N2, T1	$\mathrm{Acc_{CNN}} = 0.0016\mathrm{F2} + -0.529\mathrm{L2} + -0.0092\mathrm{L3} + -0.2635\mathrm{N1} + 0.0375\mathrm{N2} + 0.0038\mathrm{T1} + 0.9829$	0.9639	0.0199	0.0254

The fourth regression model represents the most important outcome of this study, since it combines simplicity with strong predictive accuracy. The model expresses CNN performance as a linear function of six dataset meta-features. Among these features, L2 emerges as the most influential factor, carrying the strongest negative weight (−0.529). This measure represents the error rate of a linear classifier, and its large negative coefficient indicates that datasets with weak linear separability pose significant challenges for CNNs. In other words, when classes cannot be separated by simple linear boundaries, CNN accuracy declines sharply.

The second major negative contributor is N1 (−0.2635), which quantifies the degree of class overlap using nearest-neighbor analysis. Higher N1 values indicate that samples from different classes are intermixed within the feature space, making it difficult for CNNs to identify meaningful local patterns. This result confirms that CNNs, while powerful, remain sensitive to data geometry and particularly struggle with overlapping class regions.

A smaller but still negative influence is observed with L3 (−0.0092). This feature reflects higher-order linearity errors, suggesting that nonlinear feature–class relationships marginally hinder CNN learning. Although its coefficient is much smaller than that of L2 or N1, the negative sign reinforces the idea that linear separability remains a key determinant of performance. For most features, the correlation with transformer accuracy exceeds that with CNN accuracy; however, L3 shows the opposite trend.

On the positive side, several features provide modest improvements. F2 contributes slightly (0.0016), indicating that certain statistical properties of the dataset exert a weak but supportive role in CNN performance. More important is N2 (0.0375), which measures neighborhood consistency by assessing whether samples are surrounded by neighbors of the same class. Its positive effect suggests that CNNs benefit from well-structured local patterns, as these provide clearer signals for convolutional feature extraction. Similarly, T1 (0.0038), which captures adherence subset consistency within class manifolds, adds a minor positive contribution, highlighting the advantage of regular structural organization within the dataset.

Finally, the intercept of 0.9829 points to a strong baseline performance across datasets. This high constant suggests that CNNs inherently maintain a strong level of predictive accuracy, even before the influence of dataset-specific characteristics is considered. The negative effects of L2, N1, and L3 reduce this baseline under difficult conditions, while the positive contributions of F2, N2, and T1 help preserve or enhance performance when the dataset exhibits a favorable structure. All these values collectively contribute to accurately estimating the true accuracy values.

Overall, this regression model achieves a correlation coefficient of 0.9639, with low MAE (0.0199) and RMSE (0.0254). These values are nearly identical to those of the third regression model but are obtained with a more compact set of features. This makes the fourth model not only statistically powerful but also interpretable, as it clearly identifies the dataset properties that matter most for CNNs. The result underscores the dominant role of linear separability and class overlap in shaping CNN performance, while also acknowledging the supportive effects of neighborhood structure and adherence subsets.

4.3 | Transformer Accuracy Estimation With Meta-Features

In our study, we also aimed to understand the relationship between the meta-parameters of the dataset and the predictive accuracy of the transformer classifier. We determined the statistical and predictive significance of each meta-parameter in terms of transformer accuracy using various feature importance and correlation analysis techniques. We applied the same techniques used for CNNs to analyze the relationships between meta-features and transformer accuracy, and the results are shown in Table 6.

Table 6 presents the relationship between various dataset-level meta-features and classification accuracy, as evaluated using multiple importance and statistical metrics. Several complexity measures (particularly N1, L2, and N3) demonstrate strong negative correlations with classification accuracy ($r = -0.74$, -0.68, and -0.69, respectively), indicating that datasets characterized by high instance overlap and class boundary complexity tend to result in lower classifier performance. These features also show high RF instance scores and highly significant F scores (e.g., N1: $F = 311.6$, $p < 0.001$), confirming their relevance in capturing factors influencing model success.

In contrast, F3 and F4, which are typically related to class separability or distributional characteristics, exhibit strong positive correlations with accuracy ($r = 0.55$ and 0.44, respectively) and highly significant univariate F scores ($F = 116.2$ and 63.7, $p < 0.001$). These findings suggest that these features may positively influence the ease of learning in classification tasks. Other features, such as label entropy and T1, also show moderate to strong negative correlations, implying that label

TABLE 6 | Significant meta-features influencing transformer accuracy ($p < 0.05$).

Feature	Random forest importance	MI (nonlinear relationship)	Correlation of accuracy (r)	Univariate feature selection	
				F score	p value
F1v	0.0269	0.2521	0.1894	9.82	0.0019
F2	0.0058	0.1710	−0.2401	16.16	0.0001
F3	0.0121	0.2980	0.5529	116.21	0.0000
F4	0.0099	0.2475	0.4409	63.70	0.0000
L1	0.0121	0.3014	−0.2639	19.76	0.0000
L2	0.1591	0.6427	−0.6768	223.15	0.0000
L3	0.0257	0.2064	−0.1816	9.00	0.0030
H (D)	0.0375	0.4164	−0.3643	40.41	0.0000
N1	0.4716	0.6821	−0.7358	311.60	0.0000
N2	0.0173	0.3785	−0.6040	151.61	0.0000
N3	0.0589	0.6549	−0.6923	242.97	0.0000
N4	0.0249	0.3357	−0.4791	78.63	0.0000
T1	0.0061	0.1481	−0.3166	29.42	0.0000

distribution and class imbalance contribute to predictive difficulty. By applying linear regression on the meta-features and transformer accuracy results, we generated the models shown in Table 7.

The first regression formula for the transformer shows that $H(D)$, average FC, linear classifier error (L2), and fraction of points over class boundaries (N1) have negative coefficients, which means that higher disorder, correlated features, poor linear separability, and overlapping classes reduce transformer accuracy. In contrast, the 1NN leave-one-out error (N3) contributes positively, suggesting that transformers benefit from datasets with complex local structures.

The second formula identifies neighborhood complexity (N3) as the main limiting factor, with additional negative effects from linear classifier errors (L3, L1) and overlap-related measures (N2) as well as entropy (H). Overall, this model shows that transformers are vulnerable when local irregularities and disorder dominate, although it still achieves good predictive power (corr. coef.: 0.9041).

The third one expands the feature set and provides a more nuanced view. It highlights class overlap (N1) as the strongest negative contributor, while N3 switches to a positive role, indicating that some degree of local structure benefits transformers. Despite including many features with relatively small effects, this model achieves a higher correlation (0.9329), suggesting that balancing both global and local dataset descriptors improves predictive accuracy.

The fourth model provides a concise yet powerful explanation of how dataset characteristics influence predictive accuracy. The most influential factor here is N1 (−0.3854), which measures class overlap in the nearest-neighbor sense. Its strong negative effect indicates that transformers, despite their global self-attention mechanism, struggle significantly when different classes are intermixed in the feature space. This confirms that overlapping signals remain a fundamental barrier to accurate classification.

Another important negative contributor is L2 (−0.2059), which represents the error rate of a linear classifier. Its substantial weight highlights the difficulty transformers face when data are not linearly separable, reinforcing the idea that clarity of decision boundaries is critical to their performance. Besides, a smaller negative contribution comes from T1 (−0.0303), representing adherence subset consistency, which suggests that overly rigid dataset structures may hinder the flexibility of self-attention mechanisms.

On the positive side, N2 (0.0124) provides a modest improvement. Since N2 quantifies neighborhood consistency, this suggests that transformers benefit when local class groupings are coherent, even though their attention mechanism is designed to exploit global dependencies. The statistical features F2 (0.0062) and F3 (0.0001) also contribute weakly in the positive direction, indicating that variance and distributional properties of the features slightly support model learning, although their impact is minimal compared to structural factors.

The intercept term (1.012) reflects a high baseline accuracy, underscoring that transformers are capable of achieving strong predictive results across diverse datasets. However, this baseline is notably pulled downward by class overlap (N1) and weak linear separability (L2).

Overall, this regression model highlights that while transformers can leverage both global and local information, they are still fundamentally constrained by the dataset geometry. Class overlap and linear inseparability remain the strongest barriers, while statistical features and neighborhood consistency provide only incremental gains.

4.4 | Comparison of CNN and Transformer Models on Dataset Complexity Measures

We can summarize the meta-feature–classifier relations according to their similarities and differences as below: An in-depth comparison of meta-FCs reveals that both CNNs and transformers benefit most from datasets characterized by strong

TABLE 7 | Equations for transformer accuracy with different meta-features.

Input parameters	Formula for transformer accuracy with selected parameters	Corr. coef.	MAE	RMSE
All	ZeroRule	−0.1375	0.077	0.0947
All	$\mathrm{Acc_T} = -0.0129\mathrm{H} + -0.025\mathrm{L1} + -0.0962\mathrm{L3} + -0.0553\mathrm{N2} + -0.4761\mathrm{N3} + 1.0253$	0.9041	0.0307	0.0405
Only parameters with $p < 0.05$	$\mathrm{Acc_T} = -0.0113\mathrm{H} + -0.0043\mathrm{F2} + -0.0155\mathrm{F3} + -0.0034\mathrm{F4} + -0.0131\mathrm{L1} + -0.0944\mathrm{L2} + -0.0242\mathrm{L3} + -0.7185\mathrm{N1} + 0.0211\mathrm{N2} + 0.4501\mathrm{N3} + -0.089\mathrm{N4} + -0.0224\mathrm{T1} + 1.0415$	0.9329	0.0266	0.0341
F2, F3, L2, N1, N2, T1	$\mathrm{Acc_T} = 0.0062\mathrm{F2} + 0.0001\mathrm{F3} + -0.2059\mathrm{L2} + -0.3854\mathrm{N1} + 0.0124\mathrm{N2} + -0.0303\mathrm{T1} + 1.012$	0.922	0.0272	0.0366

Note: The regression models indicate how transformer accuracy is influenced by dataset-level meta-features.

feature discriminability. As seen in Figure 7(a), features such as F3 (maximum feature efficiency), F4 (collective feature efficiency), and F1v (directional separability) exhibit the highest positive correlations with accuracy across both models. These results suggest that when features offer clear class separation, either individually or collectively, both architectures are better able to learn and generalize. CNNs also show modest positive correlation with F1, MMI, and MaxMI, although their impact is statistically weaker, indicating a secondary role for global separability and information-based attributes.

Negative correlations are consistently observed for complexity-related meta-features such as N1–N4, L2, and L3, which quantify class boundary complexity, nearest-neighbor confusion, and linear separability. These features significantly hinder both models, with transformers generally showing greater sensitivity to nonlinearity-related features such as L3, L1, $H(D)$, and F2, implying performance degradation in the presence of overlapping or irregular class boundaries. In contrast, CNNs appear more resilient to some of these global complexities but are still adversely affected when local decision boundaries become less distinct.

Some features, including T2, MMI, MaxMI, and dataset size indicators, exhibit a negligible correlation with model accuracy, implying limited predictive utility. This suggests that both architectures are relatively robust to dataset density and general statistical information when isolated from structural context. Collectively, the findings underscore that feature separability is a key driver of performance, whereas nonlinearity and boundary complexity impair learning, particularly for transformers. These patterns reflect architectural differences: CNNs leverage local, spatially consistent patterns, while transformers rely more on capturing global data relationships.

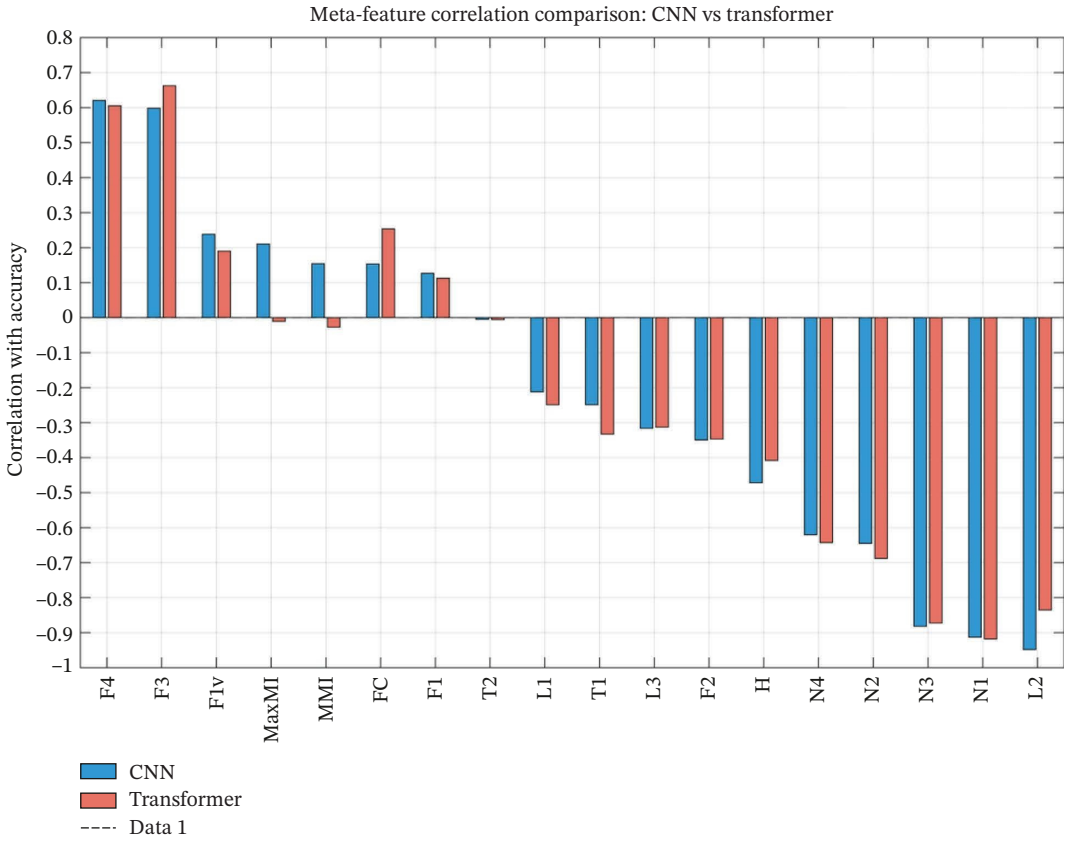
4.4.1 | Commonalities Across CNN and Transformer

A comparative analysis reveals that certain meta-features consistently influence both CNN and transformer performance, albeit to varying degrees. L2 (training error of a linear classifier) emerges as the most dominant feature across both models, highlighting the foundational role of linear separability in DL performance. N1, N3, and N4 quantify class boundary complexity through graph-based and neighborhood-based methods. Also, it shows strong negative correlations with accuracy and high MI values, as seen in Figure 7(b), indicating that datasets with more complex, overlapping boundaries are harder for both models. Similarly, H (entropy) is moderately negatively correlated with performance, suggesting that higher class uncertainty reduces model accuracy. While F3 (individual feature efficiency) exhibits a positive linear correlation in both cases, it is slightly more predictive for transformers, implying that transformers benefit more from well-separated features. In contrast, T1 (topological manifold adherence) has a weak but consistent negative effect, supporting that datasets with irregular manifold structures introduce additional complexity for both models. Overall, these common behaviors emphasize the importance of class separability, feature overlap, and decision boundary complexity as shared drivers of deep model performance.

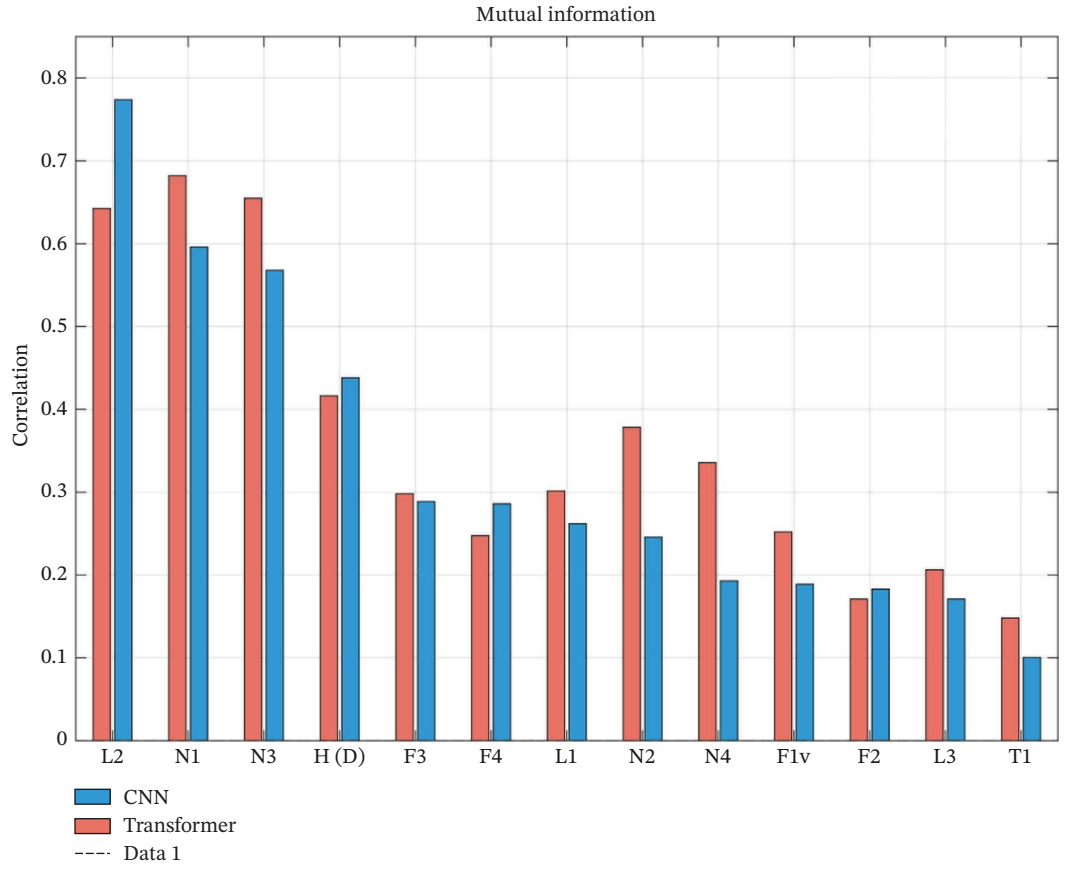
4.4.2 | Differences Between CNN and Transformer

The comparative analysis of meta-feature relationships with CNN and transformer performance reveals insights into how different structural and geometric aspects of datasets affect each model:

1. L2 (linear classifier training error) is the most influential meta-feature for both models, highlighting that basic linear separability remains a critical factor. However, its stronger influence on CNNs suggests that CNNs are more sensitive to linear decision boundaries, possibly due to their local receptive fields and hierarchical structure.
2. The stronger correlations of N1, N2, N3, N4, T1, F1v, F3, and L1 with transformer accuracy imply that transformers better exploit complex geometric and topological structures. These features quantify class boundary complexity (N1–N4), manifold adherence (T1), and directional class separability (F1v, F3, L1), indicating that transformers benefit more from datasets with rich, structured feature spaces and nonlinear separability.
3. In the case of F2, F3, and H (entropy), although linear correlations with transformer accuracy are higher, their nonlinear relationships are stronger with CNNs. This suggests that CNNs may leverage these characteristics



(a)



(b)

FIGURE 7 | Meta-feature correlation comparison and MI of CNNs and transformer.

through complex internal representations even when they do not linearly align with performance.

4. Conversely, L3 (nonlinearity of a linear classifier) shows a stronger nonlinear relationship with transformer performance, despite a weaker linear correlation. This implies that transformers are more capable of modeling nonlinear class boundaries, aligning with their ability to capture long-range dependencies and global context.

5 | Discussion and Future Work

This study presents a systematic meta-learning analysis of two state-of-the-art DL architectures (CNNs and transformers) on static, tabular, and numerical datasets. Unlike traditional benchmarking, the objective was to understand how intrinsic dataset characteristics, captured through statistical, geometric, and structural meta-features, influence the predictive behavior and sensitivity of these models.

To preserve architectural neutrality, both models were evaluated using standard hyperparameters rather than fine-tuning for maximum accuracy. This enabled clearer interpretation of their inherent learning behaviors and their interaction with dataset properties. The results show that both architectures benefit from datasets with high feature discriminability, as indicated by strong positive correlations with meta-features like F3 and F4, which capture local and global class separability. Conversely, performance deteriorates on datasets exhibiting complex class boundaries and high nonlinearity, reflected by features such as L2, N1–N4, and L3. Transformers are especially sensitive to information-theoretic properties such as MI and feature entropy, while CNNs are more influenced by local boundary complexity. These distinctions reflect their underlying mechanisms; CNNs prioritize local patterns, whereas transformers capture global dependencies.

By identifying meta-features that either enhance or hinder performance, this study deepens the understanding of DL behavior on structured, nonsequential data. It highlights the importance of model-aware data analysis: selecting CNNs or transformers without assessing data geometry and distribution may lead to suboptimal performance and inefficient resource usage. The deliberate focus on simple, well-characterized datasets enabled a controlled evaluation environment.

Although metrics such as MCC, F1 score, AUC, or ROC [38] could offer additional insights, particularly for imbalanced datasets, the balanced nature of the data in this study justified the exclusive use of accuracy. The study's aim was interpretability, not optimization. As such, the choice of accuracy and the avoidance of hyperparameter tuning align with the meta-analytical framework.

This work contributes in two primary ways: first, by characterizing the training and inference behavior of CNNs and transformers through the lens of meta-features; and second, by proposing a predictive framework to estimate their performance on unseen datasets. This approach facilitates more informed, interpretable, and resource-efficient model selection in practical applications.

Future studies may extend the proposed meta-learning framework to more challenging scenarios, including imbalanced and multiclass datasets, to assess the generalizability of the observed meta-feature–performance relationships. In addition, integrating post hoc explainability methods such as SHAP or LIME could support a more detailed interpretation of model decisions and their sensitivity to dataset characteristics. Exploring scalable and interpretable learning strategies within this context, as discussed in the recent literature, represents a natural continuation of this work [39].

6 | Conclusions

This study presents a meta-learning framework aimed at understanding how DL architectures, specifically 1D-CNNs and transformers, behave on static, tabular, and numerical datasets. By analyzing 296 datasets through 17 structural, geometric, and statistical meta-features, the study identifies how intrinsic dataset characteristics influence model performance, without relying on extensive training or fine-tuning. Instead of focusing solely on prediction accuracy, the work emphasizes model sensitivity to dataset properties and provides interpretability into their internal learning behaviors.

The results demonstrate that both CNNs and transformers benefit significantly from datasets with strong feature separability. Meta-features such as F3 (maximum feature efficiency) and F4 (collective feature efficiency) show consistently strong positive correlations with accuracy, indicating that well-separated class distributions enhance performance across both architectures. Conversely, features such as L2, N1–N4, and L3, which capture class boundary complexity and nonlinearity, are negatively correlated, especially for CNNs, whose performance degrades more sharply in complex, overlapping data spaces. Transformers show stronger sensitivity to information-theoretic measures such as MI (MMI, MaxMI) and entropy, while CNNs are more influenced by local geometrical features. These differences highlight the distinct inductive biases of the models: Transformers leverage global statistical dependencies via self-attention, while CNNs prioritize local spatial consistency and class boundaries.

The primary contribution lies in demonstrating that meta-features alone can predict the relative performance of deep models across datasets, eliminating the need for exhaustive empirical evaluations. This not only enhances interpretability and scalability but also promotes sustainable machine learning practices by reducing unnecessary computation. Furthermore, the findings emphasize the importance of data-aware model selection: understanding the geometry and complexity of datasets before model deployment can prevent mismatches between algorithm and data, thus improving efficiency and predictive accuracy.

Author Contributions

Faruk Bulut: responsible for conceptualization, project administration, methodology, dataset collection, preprocessing, data analysis, code implementation, draft preparation, and manuscript revision.

İknur Dönmez: responsible for conceptualization, methodology, code implementation, result generation, statistical analysis, interpretation, visualization, model implementation, and manuscript revision.

Acknowledgments

This research was supported by the University of Essex and its laboratory facilities, which provided access to high-performance big data servers.

Artificial intelligence tools, specifically ChatGPT (OpenAI, 2025), were used solely to improve the clarity and readability of the manuscript. No content, analysis, or results were generated by AI tools.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: the open access publication of which was funded by the University of Essex.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

C/C++, Python, and MATLAB programming languages are used to perform this study. All source codes with documentations, benchmark datasets, output files, complete experimental results with different metrics, and full comparative tables are publicly accessible through the provided URL for evaluation, review, and further research: https://sites.google.com/site/bulutfaruk/study-of-meta-learning.

References

1. A. Vettoruzzo, M. R. Bouguelia, J. Vanschoren, T. Rögnvaldsson, and K. C. Santosh, "Advances and Challenges in Meta-Learning: A Technical Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, no. 7 (2024): 4763–4779, https://doi.org/10.1109/tpami.2024.3357847.

2. A. Rivolli, L. P. Garcia, C. Soares, J. Vanschoren, and A. C. de Carvalho, "Meta-Features for Meta-Learning," *Knowledge-Based Systems* 240 (2022): 108101, https://doi.org/10.1016/j.knosys.2021.108101.

3. T. Ruhkopf, A. Mohan, D. Deng, A. Tornede, F. Hutter, and M. Lindauer, "Masif: Meta-Learned Algorithm Selection Using Implicit Fidelity Information," *Transactions on Machine Learning Research* (2023).

4. H. Chen and J. Bajorath, "Meta-Learning for Transformer-Based Prediction of Potent Compounds," *Scientific Reports* 13, no. 1 (2023): 16145, https://doi.org/10.1038/s41598-023-43046-5.

5. W. Zhao, Y. Li, H. Wang, and H. Lu, "First-Order Cross-Domain Meta Learning for Few-Shot Remote Sensing Object Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2026).

6. K. Surendro, M. I. C. Rachmatullah, and J. Santoso, "Improving 1d Convolutional Neural Network (1D Cnn) Performance in Processing Tabular Datasets Using Principal Component Analysis" (2022).

7. Z. Zhang, J. Yang, Y. Ai, and F. Luan, "CTMNR: A Hybrid CNN-Transformer Approach for Enhanced Multi-Nuclide Classification in Radiation Spectroscopy," *IEEE Transactions on Nuclear Science* (2025): https://ieeexplore.ieee.org/abstract/document/11348984.

8. N. Hollmann, S. Müller, L. Purucker, et al., "Accurate Predictions on Small Data With a Tabular Foundation Model," *Nature* 637, no. 8045 (2025): 319–326, https://doi.org/10.1038/s41586-024-08328-6.

9. J. Jang, J. Pyo, Y. I. Yoon, and J. Choi, "Meta-Transformer: A Meta-Learning Framework for Scalable Automatic Modulation Classification," *IEEE Access* 12 (2024): 9267–9276, https://doi.org/10.1109/access.2024.3352634.

10. R. Alkanhel, E. S. M. El-kenawy, A. A. Abdelhamid, et al., "Network Intrusion Detection Based on Feature Selection and Hybrid Metaheuristic Optimization," *Computers, Materials and Continua* 74, no. 2 (2023).

11. E. S. M. El-Kenawy, N. Khodadadi, S. Mirjalili, et al., "Metaheuristic Optimization for Improving Weed Detection in Wheat Images Captured by Drones," *Mathematics* 10, no. 23 (2022): 4421, https://doi.org/10.3390/math10234421.

12. J. Eberlein, D. Rodriguez, and R. Harrison, "The Effect of Data Complexity on Classifier Performance," *Empirical Software Engineering* 30, no. 1 (2025): 16, https://doi.org/10.1007/s10664-024-10554-5.

13. M. Gyanchandani, R. Wadhvani, and S. Shukla, "Data Complexity Measures for Classification of a Multi-Concept Dataset," *Multimedia Tools and Applications* 84, no. 2 (2025): 571–602.

14. T. K. Ho and M. Basu, "Complexity Measures of Supervised Classification Problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, no. 3 (2002): 289–300, https://doi.org/10.1109/34.990132.

15. M. Han, H. Guo, and W. Wang, "A New Data Complexity Measure for Multi-Class Imbalanced Classification Tasks," *Pattern Recognition* 157 (2025): 110881, https://doi.org/10.1016/j.patcog.2024.110881.

16. F. Kächele and N. Schneider, "Cluster Validation Based on Fisher's Linear Discriminant Analysis," *Journal of Classification* 42, no. 1 (2025): 54–71.

17. F. Bulut and M. F. Amasyali, "Locally Adaptive K Parameter Selection for Nearest Neighbor Classifier: One Nearest Cluster," *Pattern Analysis and Applications* 20, no. 2 (2017): 415–425, https://doi.org/10.1007/s10044-015-0504-0.

18. P. Birzhandi, K. T. Kim, and H. Y. Youn, "Reduction of Training Data for Support Vector Machine: A Survey," *Soft Computing* 26, no. 8 (2022): 3729–3742, https://doi.org/10.1007/s00500-022-06787-5.

19. H. Gong, Y. Li, J. Zhang, B. Zhang, and X. Wang, "A New Filter Feature Selection Algorithm for Classification Task by Ensembling Pearson Correlation Coefficient and Mutual Information," *Engineering Applications of Artificial Intelligence* 131 (2024): 107865, https://doi.org/10.1016/j.engappai.2024.107865.

20. Í. Santana, B. Serrano, M. Schiffer, and T. Vidal, "Support Vector Machines With the Hard-Margin Loss: Optimal Training via Combinatorial Benders' Cuts," *Journal of Global Optimization* 92, no. 1 (2025): 205–225, https://doi.org/10.1007/s10898-025-01483-8.

21. A. O. Ige and M. Sibiya, "State-of-the-Art in 1d Convolutional Neural Networks: A Survey," *IEEE Access* 12 (2024): 144082–144105, https://ieeexplore.ieee.org/abstract/document/10609371.

22. R. Kruse, S. Mostaghim, C. Borgelt, C. Braune, and M. Steinbrecher, "Multi-Layer Perceptrons," in *Computational Intelligence: A Methodological Introduction* (Cham: Springer International Publishing, 2022), 53–124.

23. L. S. Pusztaházi, G. Eigner, and O. Csiszár, "Parametric Activation Functions for Neural Networks: A Tutorial Survey," *IEEE Access* (2024).

24. A. Shehzad, F. Xia, S. Abid, et al., "Graph Transformers: A Survey," *IEEE Transactions on Neural Networks and Learning Systems* (2026).

25. J. J. Jeong and G. Koo, "Adalo: Adaptive Learning Rate Optimizer With Loss for Classification," *Information Sciences* 690 (2025): 121607, https://doi.org/10.1016/j.ins.2024.121607.

26. G. Atteia, E. S. M. El-kenawy, N. A. Samee, et al., "Adaptive Dynamic Dipper Throated Optimization for Feature Selection in Medical Data," *Computers, Materials and Continua* 75, no. 1 (2023): 1883–1900, https://doi.org/10.32604/cmc.2023.031723.

27. A. A. Abdelhamid, E. S. M. El-Kenawy, A. Ibrahim, et al., "Innovative Feature Selection Method Based on Hybrid Sine Cosine and Dipper Throated Optimization Algorithms," *IEEE Access* 11 (2023): 79750–-79776, https://doi.org/10.1109/access.2023.3298955.

28. "Kaggle Classification Datasets," (2025), https://www.kaggle.com/datasets?tags=13302-Classification.

29. K. Bache and M. Lichman, in *UCI Machine Learning Repository* (University of California, 2025), http://archive.ics.uci.edu/ml.

30. J. Alcal´a-Fdez, "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework," *Journal of Multiple-Valued Logic and Soft Computing* 17, no. 2-3 (2011): 255–287.

31. A. Orriols-Puig, N. Macia, and T. K. Ho, "Documentation for the Data Complexity Library in C++," *Universitat Ramon Llull, La Salle* 196, no. 1-40 (2010): 12.

32. E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. (Morgan Kaufmann, 2016).

33. G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, "Linear Regression," in *An Introduction to Statistical Learning: With Applications in Python* (Cham: Springer international publishing, 2023), 69–134.

34. G. Biau and E. Scornet, "A Random Forest Guided Tour," *Test* 25, no. 2 (2016): 197–227, https://doi.org/10.1007/s11749-016-0481-7.

35. P. Stoica and P. Babu, "Pearson–Matthews Correlation Coefficients for Binary and Multinary Classification," *Signal Processing* 222 (2024): 109511, https://doi.org/10.1016/j.sigpro.2024.109511.

36. J. A. Warwicker and S. Rebennack, "Efficient Continuous Piecewise Linear Regression for Linearising Univariate Non-Linear Functions," *IISE Transactions* 57, no. 3 (2025): 231–245, https://doi.org/10.1080/24725854.2023.2299809.

37. B. Gregorutti, B. Michel, and P. Saint-Pierre, "Correlation and Variable Importance in Random Forests," *Statistics and Computing* 27, no. 3 (2017): 659–678, https://doi.org/10.1007/s11222-016-9646-1.

38. D. Chicco and G. Jurman, "The Matthews Correlation Coefficient (MCC) Should Replace the ROC AUC as the Standard Metric for Assessing Binary Classification," *BioData Mining* 16, no. 1 (2023): 4, https://doi.org/10.1186/s13040-023-00322-4.

39. H. Myriam, A. A. Abdelhamid, E. S. M. El-Kenawy, et al., "Advanced Meta-Heuristic Algorithm Based on Particle Swarm and Al-Biruni Earth Radius Optimization Methods for Oral Cancer Detection," *IEEE Access* 11 (2023): 23681–23700, https://doi.org/10.1109/access.2023.3253430.