

Article

Designing Channel Attention Fully Convolutional Networks with Neural Architecture Search for Customer Socio-Demographic Information Identification Using Smart Meter Data

Zhirui Luo ¹, Qingqing Li ², Ruobin Qi ³ and Jun Zheng ^{1,*}

¹ Department of Computer Science and Engineering, New Mexico Institute of Mining and Technology, Socorro, NM 87801, USA; zhirui.luo@student.nmt.edu

² Department of Computer and Information Sciences, Towson University, Towson, MD 21252, USA; qingqingli@towson.edu

³ Department of Computer Science, California State University Northridge, Northridge, CA 91330, USA; ruobin.qi@csun.edu

* Correspondence: jun.zheng@nmt.edu; Tel.: +1-575-835-6182

Abstract: Background: Accurately identifying the socio-demographic information of customers is crucial for utilities. It enables them to efficiently deliver personalized energy services and manage distribution networks. In recent years, machine learning-based data-driven methods have gained popularity compared to traditional survey-based approaches, owing to their time and cost efficiency, as well as the availability of a large amount of high-frequency smart meter data. **Methods:** In this paper, we propose a new method that harnesses the power of neural architecture search to automatically design deep neural network architectures tailored for identifying various socio-demographic information of customers using smart meter data. We designed a search space based on a novel channel attention fully convolutional network architecture. Furthermore, we developed a search algorithm based on Bayesian optimization to effectively explore the space and identify high-performing architectures. **Results:** The performance of the proposed method was evaluated and compared with a set of machine learning and deep learning baseline methods using a smart meter dataset widely used in this research area. Our results show that the deep neural network architectures designed automatically by our proposed method significantly outperform all baseline methods in addressing the socio-demographic questions investigated in our study.

Keywords: neural architecture search; fully convolutional network; attention; Bayesian optimization; socio-demographic information identification; smart meter



Academic Editor: Ke-Lin Du

Received: 11 November 2024

Revised: 26 December 2024

Accepted: 27 December 2024

Published: 10 January 2025

Citation: Luo, Z.; Li, Q.; Qi, R.; Zheng, J. Designing Channel Attention Fully Convolutional Networks with Neural Architecture Search for Customer Socio-Demographic Information Identification Using Smart Meter Data. *AI* **2025**, *6*, 9. <https://doi.org/10.3390/ai6010009>

Copyright: © 2025 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

Socio-demographic information of customers is important for promoting the quality of policy-making, marketing, and system operation of smart grids. For instance, utility retailers can utilize this information to make informed decisions regarding demand response and energy efficiency programs. Previous studies have shown that socio-demographic characteristics of households can impact appliance usage [1]. Understanding the socio-demographic information of households can also improve the accuracy of district load forecasting [2]. Based on the socio-demographic information, energy affordability programs and policy-makers can identify the causes of the high energy burden on low-income households and adjust schedules and policies accordingly to alleviate their burden [3,4].

Traditionally, surveys have been employed to obtain customers' socio-demographic information, but they are time-consuming and cost-inefficient. However, the power consumption pattern can be significantly influenced by the social and economic characteristics of consumers. The development of Advanced Metering Infrastructure (AMI) enables the collection of large amounts of real-time electricity consumption data, leading to a surge in data-driven methods in recent years for inferring socio-demographic information from power consumption data [5–14].

Typical data-driven socio-demographic information identification methods involve feature extraction, feature selection, and classification or regression. For example, Beckel et al. [5] proposed a household characteristic estimation system named CLASS, which extracts and selects features and identifies household characteristics based on them. Beckel et al. [6] extended the previous work with regression to estimate continuous values of household characteristics through feature extraction and selection. Hopf et al. [7] proposed an enhanced CLASS system, named eClass, which quadruples the number of features of the CLASS system and applies feature filtering methods to improve the performance of classification. Fahim and Sillitti [10] developed a framework to infer household characteristics that extracts features from time-series power consumption data and utilizes Random Forest (RF) for classification. In [9], the Grade Correspondence Analysis (GCA) with posterior clustering and visualization was applied to extract important features from power consumption data and feed the features into three different classifiers, including artificial neural network (ANN), k-nearest neighbor (kNN), and support vector machine (SVM), to identify household characteristics. Yan et al. [11] proposed an efficient approach to identify household characteristics using smart meter data by combining time–frequency features and employing the RF algorithm for feature selection, followed by an SVM classifier, leading to improved performance compared to conventional methods.

Recently, deep learning has emerged as a promising solution for automatically extracting discriminative information from time-series data, including power consumption data [15,16]. Several studies have explored the potential of deep learning in data-driven socio-demographic information identification. For example, Wang et al. [8] developed a convolutional neural network (CNN) to automate the feature extraction process for socio-demographic information identification, leveraging the abundance of fine-grained real-time electricity consumption data. In a study by Wang et al. [13], CNN models based on either 1D or 2D convolution operations were investigated for extracting features from smart meter data to identify the socio-demographic information of consumers. Additionally, Xu et al. [14] proposed a deep model that combines the strengths of CNNs and long short-term memory (LSTM) to capture temporal and spatial features from power consumption data for socio-demographic information identification. These works demonstrate the effectiveness of deep learning-based methods in extracting discriminative features when compared to traditional manual feature engineering-based methods.

One of the primary limitations of the current deep learning-based methods used for data-driven socio-demographic information identification lies in their dependence on manually designed deep neural network (DNN) architectures, which limits their capabilities in capturing essential temporal and spatial patterns required to accurately identify diverse socio-demographic information. To overcome this drawback, we propose a novel approach in this paper, which harnesses the power of neural architecture search (NAS) to automatically discover DNN architectures specifically tailored for the identification of various socio-demographic information. NAS is an automated machine learning (AutoML) technique that automates the design process to find robust and high-performance DNN architectures for various science and engineering applications [17–20]. We name a discovered

DNN architecture SEACAT-Net, which is **searched** from a search space designed based on a novel **channel attention**-based fully convolutional **network** (CAFCN) architecture.

Our research makes three main contributions. Firstly, we designed a search space using a novel CAFCN architecture, enabling effective capture of both temporal and spatial patterns present in smart meter data for socio-demographic information identification. Secondly, we developed a Bayesian optimization (BO)-based search strategy to automatically explore and select the high-performing architecture for a given socio-demographic information inquiry. Lastly, we evaluated the performance of our proposed method using a widely used smart meter dataset and compared it against a set of state-of-the-art baselines, demonstrating its superior performance in data-driven socio-demographic information identification.

The rest of this paper is organized as follows: Section 2 presents the proposed NAS-based method for discovering high-performing SEACAT-Net architectures, encompassing search space design, a BO-based search strategy, and an architecture evaluation strategy. Section 3 first describes the smart meter dataset used in our experiments, followed by implementation details and baseline models for performance comparison. In Section 4, the experimental results are presented to demonstrate the effectiveness and superior performance of the proposed method. Finally, we conclude this paper in Section 5.

2. Methodology

The workflow of the proposed NAS-based method for data-driven customer socio-demographic information identification is illustrated in Figure 1, consisting of three main components: search space design, BO-based search strategy, and architecture evaluation strategy. We describe each of the three components in detail in the following sections.

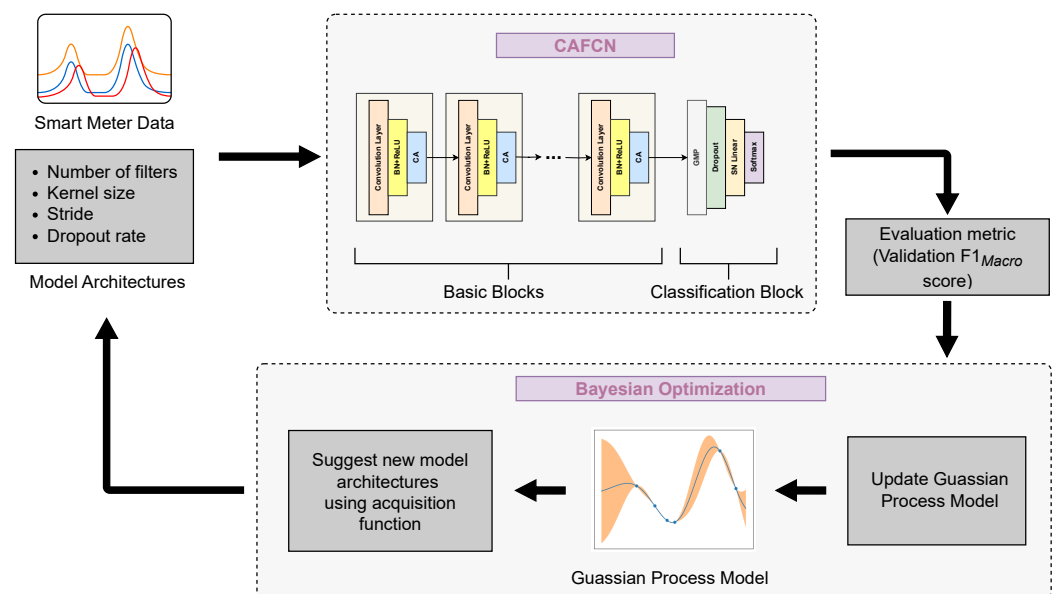


Figure 1. Workflow of the proposed NAS-based method for data-driven customer socio-demographic information identification.

2.1. Search Space Design

FCN has demonstrated its great advantage as a feature extractor for various TSC tasks [16,19,21,22]. Additionally, attention mechanisms have proven effective in enhancing the performance of different applications by focusing on important features [16,23–26]. Consequently, we designed a search space for the proposed NAS-based method based on a novel CAFCN architecture. The architecture consists of a series of sequentially stacked basic blocks, followed by a classification block as shown in Figure 2. The rationale behind

the design is to consider both architectural diversity and hyperparameter selection for a specific layer, while ensuring a reasonable search space size.

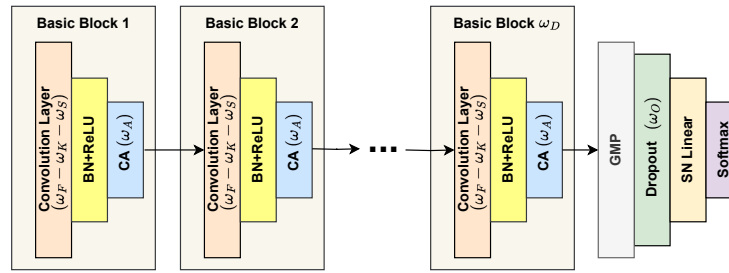


Figure 2. CAFCN architecture.

The search space of the CAFCN architecture is divided into three parts: network depth, basic block, and classification block. The network depth (n_D) denotes the number of basic blocks, ranging from 3 to 5. Each basic block sequentially stacks a convolutional (Conv) layer, a Batch Normalization (BN) layer, a ReLU activation function, and an optional channel attention (CA) layer (ω_A). Specifically, the Conv layer uses a two-dimensional squared kernel (k, k) with padding $(k - 1)/2$. The searchable hyperparameters of the Conv layer include the number of filters (n_F), kernel size (n_K), and stride size (n_S), with their possible values listed in Table 1, which reflects popular choices in CNN model design.

The adopted channel attention mechanism is the squeeze-and-excitation (SE) technique, a simple and effective approach proposed in [27]. Figure 3 provides an illustration of an SE block. Global average pooling (GAP) was initially applied as the squeeze operation to the input, extracting channel-wise statistics from it. Subsequently, the excitation operation was performed on the output of the squeeze operation, consisting of two fully connected (FC) layers, a ReLU function, and a Sigmoid activation. The output of the excitation operation represents the weights assigned to different input channels, which were then multiplied with the input to enhance the importance of specific channels. In a basic block, the SE-based CA layer is optional ($\omega_A = 0$ or 1).

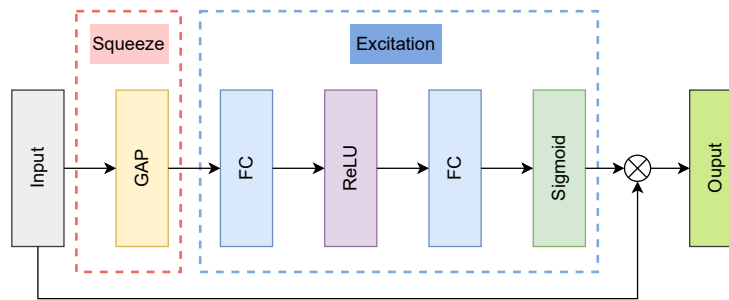


Figure 3. Operation flow of an SE block.

The only searchable hyperparameter of the classification block is the dropout rate (p_O), which is employed to prevent overfitting. Additionally, the classification block includes a global max-pooling (GMP) layer after the last basic block. We used spectrum normalization (SN) to stabilize the linear layer, followed by a softmax function for the output.

Due to the existence of over 3.9 million valid architectures in the designed search space, it is practically infeasible to perform a brute-force search through all of them to find the optimal one. Therefore, we adopted a BO-based search strategy to address this challenge.

Table 1. Search space designed for the proposed method.

Components	Hyperparameters (Ω)	Choices
Network Depth	Number of basic blocks (n_D)	3/4/5
Basic Block	Conv—number of filters (n_F)	32/64/128
	Conv—kernel size (n_K)	3/5/7/9
	Conv—stride size (n_S)	1/2/4
	CA layer (ω_A)	0/1
Classification Block	Dropout rate (p_O)	0.5/0.6

2.2. BO-Based Search Strategy

The goal of NAS is to search for a neural network architecture and/or a set of hyperparameters optimized based on an objective. BO is an effective method to achieve this goal in a designed search space. It involves building a cheap surrogate function for an objective and considering uncertainty through models like Gaussian process regression [28]. The BO-based search strategy can be formally formulated as follows:

$$\mathbf{a}^* = \arg \min_{\mathbf{a} \in \mathcal{A}} f(\mathbf{a}) \quad (1)$$

where $\mathcal{A}$ is the search space of network architectures, $f(\cdot)$ indicates the objective function, and $\mathbf{a}$ and $\mathbf{a}^*$ refer to an architecture within the designed search space and the architecture obtained by the search algorithm, respectively. $f(\cdot)$ is usually “expensive to evaluate” and the actual evaluation time is based on the size of the architecture, the size of the training dataset, training devices, etc. The BO-based search strategy treats any objective function $f(\cdot)$ as a black box, which makes it a good candidate for modeling complicated NAS problems. Recent works have already shown the capability of BO to help find robust and high-performing model architectures for different applications [29–31].

The BO-based search algorithm is depicted in Algorithm 1, which builds upon the Sequential Model-Based Optimization (SMBO) algorithm [32]. This algorithm comprises two important components: GPR for modeling the objective function, and an acquisition function for suggesting the subsequent architecture to evaluate. In the algorithm, $\mathbf{a}_t$ represents the suggested architecture suggested by the utility function, and y_t denotes its corresponding evaluated performance. The optimization process commences by evaluating k randomly selected architectures for a warm-up and then iteratively evaluates the suggested architectures with a total budget of T .

Algorithm 1 BO-based Search Algorithm

Require: f —objective function, $\mathcal{A}$ —search space, $\mathcal{U}$ —the acquisition function, $\mathcal{G}$ —the GPR model, k —the number of random initial samples, T —total search budget

```

1: procedure BO_SEARCH( $f, \mathcal{X}, \mathcal{A}, \mathcal{G}, k, T$ )
2:    $\mathcal{A}_{init} = \{\mathbf{a}_i | \mathbf{a}_i \in \mathcal{A} \text{ and } i = 1, 2, \dots, k\}$ . ▷ Randomly initialized architectures
3:    $\mathcal{D} \leftarrow \{f(\mathbf{a}) \text{ for all } \mathbf{a} \in \mathcal{A}_{init}\}$ 
4:   for  $i \leftarrow |\mathcal{D}|$  to  $T$  do
5:      $\mathcal{G} \leftarrow \text{FitModel}(\mathcal{D})$ 
6:      $\mathbf{a}_t = \arg \max_{\mathbf{a} \in \mathcal{A}} \mathcal{U}(\mathbf{a}, \mathcal{G}(\mathbf{a}, \mathcal{D}))$ 
7:      $y_t \leftarrow f(\mathbf{a}_t)$  ▷ Architecture evaluation
8:      $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{a}_t, y_t)\}$ 
9:   end for
10: end procedure

```

GPR is a non-parametric Bayesian approach that is employed to tackle regression problems due to its ability to capture relationships using a theoretically infinite number of parameters [33]. This technique proves particularly useful for addressing exploration–exploitation problems and effectively optimizing the search process. Meanwhile, the acquisition function plays a pivotal role in guiding the search process toward finding the optimal solution.

Table 2 illustrates three commonly used acquisition functions: the Upper Confidence Bound (UCB), Expected Improvement (EI), and Probability of Improvement (PI). However, it is essential to note that there is no guarantee that any single acquisition function will perform optimally for every objective. To address this uncertainty, Hoffman et al. [34] introduced a portfolio allocation strategy called GP-Hedge. This strategy outperforms individual acquisition functions by involving sampling from a pool of mixed acquisition functions. At each iteration, GP-Hedge probabilistically selects one of the acquisition functions. In our work, we adopted GP-Hedge, combining EI, PI, and UCB to enhance search efficiency and effectiveness.

Table 2. Three commonly used acquisition functions ($H^{+/-}$ —right-/left-continuous Heaviside step function; ReLU—rectified linear nonlinearity; α —improvement threshold; $\mu(\cdot)$ —expectation; β —trade-off parameter; $\sigma(\cdot)$ —standard deviation).

Acquisition Function	Abbr.	Formulation
Expected Improvement	EI	$\mathbb{E}_y[\max(\text{ReLU}(y - \alpha))]$
Probability of Improvement	PI	$\mathbb{E}_y[\max(H^-(y - \alpha))]$
Upper Confidence Bound	UCB	$\mathbb{E}_y[\max(\mu(y) + \sqrt{\beta}\sigma(y))]$

The search algorithm depicted in Algorithm 1 proposes and evaluates models sequentially, limiting its ability to take advantage of a parallel evaluation of architectures. In our work, we employed an asynchronous parallel algorithm to expedite the search process. This algorithm allows n evaluations to be performed simultaneously, which corresponds to the maximum number of parallel tasks. The GPR is updated asynchronously upon task completion. To accommodate parallel evaluations, UCB, EI, PI, and GP-Hedge can all be extended to generate n samples from the search space that jointly maximize the acquisition function. For example, an extension for EI is demonstrated in [28].

2.3. Architecture Evaluation Strategy

The focus of our work is to search for high-performing DNN architectures for data-driven socio-demographic information identification. Thus, the objective function $f()$ of an evaluated architecture represents the performance measure obtained from the validation dataset. A commonly used performance measure for data-driven socio-demographic information identification is the *accuracy*, calculated as shown in Equation (2), where N_c represents the number of correctly classified samples, and N is the total number of samples:

$$\text{Accuracy} = \frac{N_c}{N} \quad (2)$$

However, the data for most socio-demographic questions are highly imbalanced, rendering the *accuracy* measure inadequate, as it becomes biased towards major classes. Therefore, we adopted the *F1 score*, which is a more suitable performance measure for

imbalanced data. For a binary classification problem, the $F1$ score is calculated as the harmonic mean of precision (Pr) and recall (Re):

$$F1 = 2 \frac{Pr \times Re}{Pr + Re} \quad (3)$$

where the Pr and Re are formulated as

$$Pr = TP / (TP + FN) \quad (4)$$

$$Re = TP / (TP + FP) \quad (5)$$

where TP , FP , and FN are true positives, false positives, and false negatives, respectively.

In the case of multi-class classification problems, we employed the macro averaging of $F1$ scores ($F1_{Macro}$). It involves first calculating the $F1$ score for each class, $F1_i$ ($i = 1, 2, \dots, C$), in a One-versus-Rest setting, where C is the number of classes. Subsequently, $F1_{Macro}$ is computed by taking the arithmetic mean of all per-class $F1$ scores as shown in Equation (6).

$$F1_{Macro} = \frac{\sum_{i=1}^C F1_i}{C} \quad (6)$$

3. Experiments

In this section, we first describe the smart meter dataset used in our experiments. Next, we present the implementation details of the proposed method, implemented using PyTorch 2.0. Finally, we introduce a set of baseline machine learning and deep learning methods for performance comparison.

3.1. Smart Meter Dataset

The dataset used in our study is the Electricity Customer Behaviour Trial dataset obtained from the Irish Commission for Energy Regulation (CER) Smart Metering Project [35]. The data collection took place between 2009 and 2010, involving over 5000 Irish homes and businesses. The dataset comprises smart meter consumption data recorded every 30 min over a period of 536 days. For our experiments, we selected the first 75 weeks (525 days) of power consumption data from 4232 residential consumers. We divided the data into three sets: 60% for training, 20% for validation, and the remaining 20% for testing. Any null or continuously zero values present in the data were removed prior to analysis.

The dataset includes pre-trial and post-trial surveys. For performance evaluation, we utilized the answers from the pre-trial survey as labels. To ensure a fair comparison, we carefully selected ten socio-demographic questions based on existing work and our research interests. These selected socio-demographic questions are listed in Table 3. It can be observed that the data for the majority of the questions are highly imbalanced.

Table 3. Ten socio-demographic questions selected for our study.

Question Number	Socio-Demographic Info	Class Labels	Number of Samples
300	Age of chief income earner	Young (<35)	436
		Medium (35~65)	2819
		Senior (>65)	953
310	Chief income earner has retired or not	Yes	1285
		No	2947

Table 3. Cont.

Question Number	Socio-Demographic Info	Class Labels	Number of Samples
401	Social class of chief income earner	A or B	642
		C1 or C2	1840
		D or E	1593
410	Have children or not	Yes	1229
		No	3003
450	House type	Detached or bungalow	2189
		Semi-detached or terraced	1964
453	Age of the house	Old (≥ 30)	2151
		New (< 30)	2077
460	Number of bedrooms	Low ($=3$)	1884
		High ($=4$)	1470
		Very High (>4)	474
4704	Cooking facility type	Electrical	1272
		Not electrical	2960
4905	Energy-efficient light bulb proportion	Up to half	2746
		Three quarters or more	1486
5418	Level of education of the chief income earner	Primary/no education	534
		Secondary	1886
		Third level	1580

3.2. Implementation Details

3.2.1. NAS Hyperparameters

The number of randomly selected initial architectures k was set to 10. For each question, the search budget T was set to 150. We utilized the BO implementation (*gp_minimize*) from Scikit-Optimize version 0.9.0. However, it should be noted that Scikit-Optimize minimizes the objective function instead of maximizing it. Therefore, we used negative EI (NEI), negative PI (NPI), and lower confidence bound (LCB) in the GP-Hedge framework. The improvement threshold α for NEI and NPI was set to 0.01. In Scikit-Optimize, we set the exploration–exploitation trade-off parameter β of LCB to 1.96.

3.2.2. Network Training During Searching

During the search phase, we set the maximum epoch of training to 20 and employed early stopping with a patience value of 4 to expedite the search process. To train each model, we utilized the cross-entropy loss with the Adam optimizer and a step learning rate scheduler. Additionally, we applied label-smoothing with a value of 0.2 in conjunction with the cross-entropy loss [36]. The learning rate was set to 0.001, and the weight decay was set to 0.001. The step learning rate scheduler used a step of 7 to reduce the learning rate. Throughout the search, we used a batch size of 1024 to reduce the training time. It is worth noting that the architecture with the highest validation $F1_{Macro}$ score for each question after the search was retained as the best model.

3.3. Baseline Methods for Performance Comparison

To showcase the effectiveness of our proposed method, we conducted a comprehensive comparison with a set of baseline machine learning and deep learning methods. For the machine learning-based methods, we optimized their parameters using grid search. All deep learning-based methods employ manually designed DNN architectures. For performance

comparison, we set the maximum epoch to 50 and the batch size to 128 for SEACAT-Net and all deep learning-based baseline methods. To ensure a fair comparison, all baseline methods used raw power consumption data as input, similar to the proposed method.

3.3.1. SVM

SVM is a widely adopted supervised machine learning algorithm for data-driven socio-demographic information identification [5,8,9,11]. We employed the grid search to find the optimal parameters for SVM that yield the highest validation $F1_{Macro}$ score.

3.3.2. PCA + SVM (PS)

Principal Component Analysis (PCA) is a well-known technique for dimension reduction. In the PS method, we selected the top N PCA components obtained from the smart meter data as input features for an SVM classifier used for the purpose of identification. The optimal parameters of the SVM classifier were determined through grid search.

3.3.3. RF

RF is an ensemble learning method that combines multiple decision trees to make predictions. RF is also a popular supervised learning algorithm for data-driven socio-demographic information identification [10].

3.3.4. PCA + RF (PR)

Similar to the PS method, the PR method first employs PCA to reduce the dimensionality of the smart meter data. Then, RF is applied to classify the extracted PCA features.

3.3.5. CNN-SVM

CNN-SVM is a method proposed in [8] that utilizes SVM to classify features automatically extracted from a CNN model.

3.3.6. A 1D-CNN

The 1D-CNN is a convolutional neural network (CNN) architecture proposed in [13] that utilizes one-dimensional convolutional (Conv1D) operations to automatically extract features from time-series power consumption data.

3.3.7. A 2D-CNN

Instead of employing Conv1D operations, the 2D-CNN architecture proposed in [13] utilizes two-dimensional convolutional (Conv2D) operations to extract features from power consumption data organized in two dimensions, by days.

3.3.8. CNN-LSTM

CNN-LSTM is a deep learning model proposed in [14] that sequentially combines CNN and Bidirectional LSTM (BiLSTM) to capture both spatial and temporal features from power consumption data.

4. Results

4.1. Search Results

To evaluate the performance of the BO-based search algorithm, we analyzed the performance distributions of the evaluated architectures for the ten questions. Figure 4 visualizes these distributions using violin plots, revealing significant variations among different questions. The plots show that the performance of the evaluated architectures for each question varies widely, and the search algorithm can discover a few high-performing architectures.

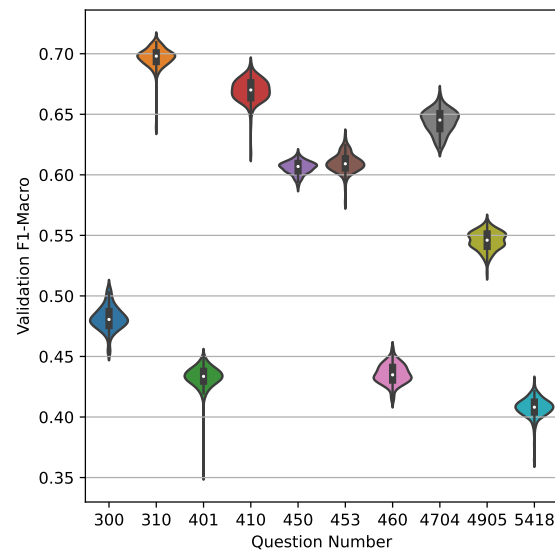


Figure 4. Distributions of validation $F1_{Macro}$ of searched architectures for the ten questions.

Additionally, we assessed the convergence of the search algorithm to determine its effectiveness. To illustrate this, we plotted the change in the cumulative maximum validation $F1_{Macro}$ score with the increment of the number of evaluated models. Figure 5 displays these plots for the ten questions. It can be observed that certain questions, such as 450, 460, and 5418, reach their maximum performance with less than 70% of the evaluated architectures. In contrast, the performance of questions like 420, 453, 4704, and 4905 peaks in the later stages of the search process. Overall, we can conclude that the BO-based search algorithm implemented in the proposed method exhibits favorable convergence within the allocated search budget.

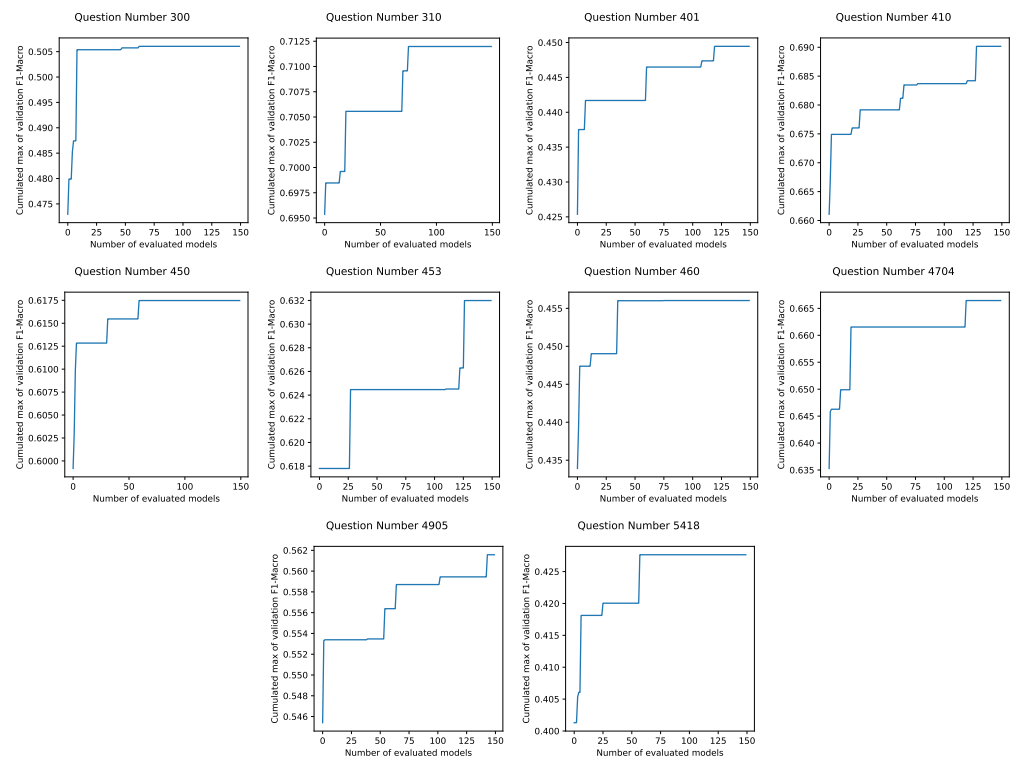


Figure 5. Maximum validation $F1_{Macro}$ score vs. number of evaluated models for the ten questions.

After the search process, we obtained the SEACAT-Net architecture for each question by selecting the one with the highest validation $F1_{Macro}$ score. Figure 6 shows the diverse SEACAT-Net architectures for all ten questions, underscoring the need for tailored DNN architectures to address various socio-demographic questions.

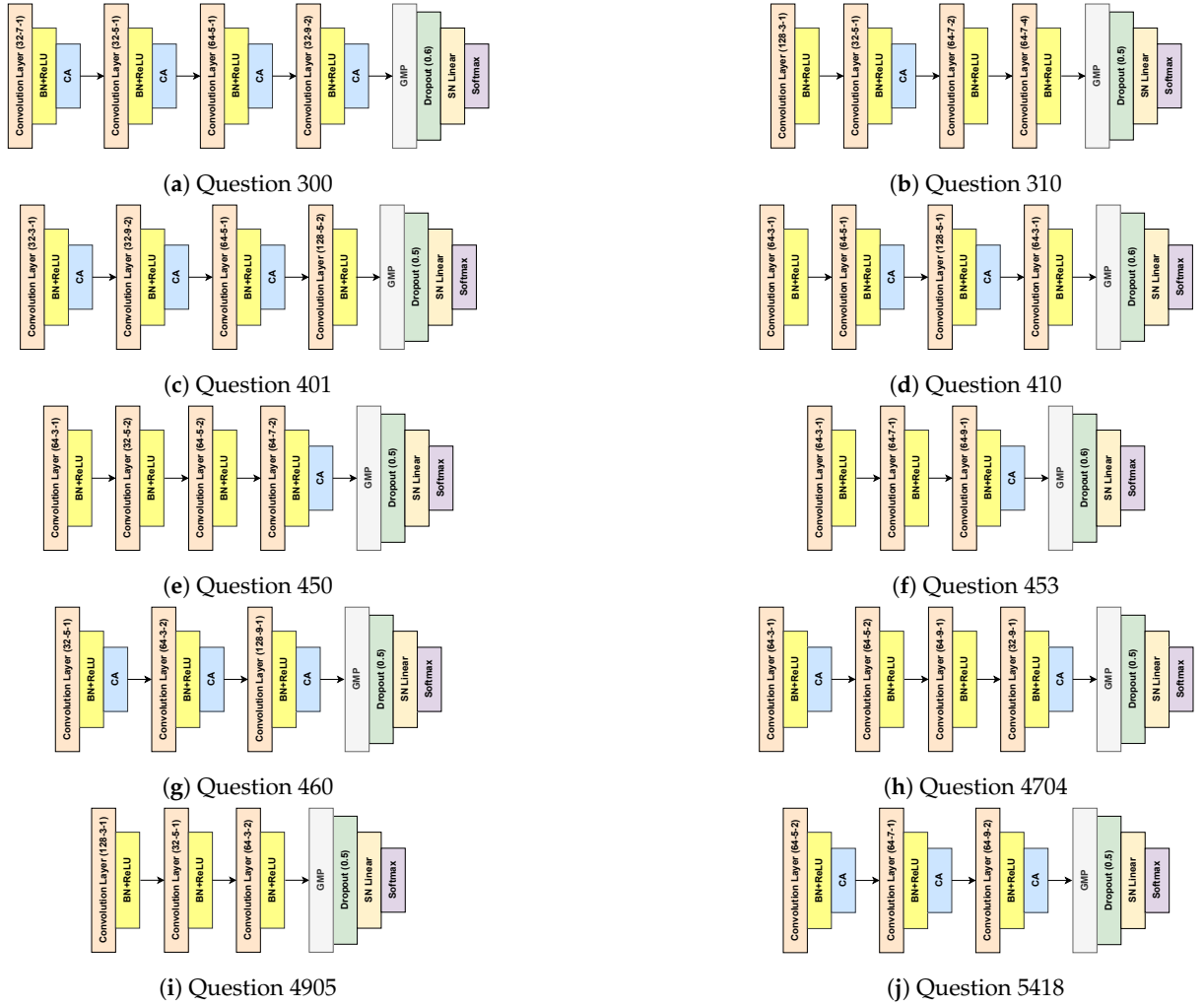


Figure 6. SEACAT-Net architectures for the ten socio-demographic questions.

4.2. Performance Comparison Results

The proposed method was compared with the set of baseline methods described in Section 3.3 using the test set. Table 4 displays the test $F1_{Macro}$ scores of the proposed method and all baselines. Among the machine learning-based baseline methods, RF achieves the highest performance in nine out of ten questions, while SVM exhibits significantly lower performance than RF. The application of PCA slightly enhances the performance of SVM, but it does not yield improvements for RF. This can be attributed to the loss of information caused by PCA, diminishing the benefits of ensemble methods like RF. Among the deep learning-based baseline methods, CNN-LSTM is the best-performing one. LSTM can effectively capture both short-term and long-term dependencies in time-series [37,38]. Therefore, combining CNN and LSTM allows for the extraction of essential spatial and temporal information from the power consumption data for the task. The results demonstrate that SEACAT-Net significantly outperforms all machine learning and deep learning-based baseline methods for the ten questions, highlighting the capability of the proposed method to discover high-performing channel attention-based DNN architectures tailored to specific socio-demographic questions.

Table 4. Performance comparison in terms of test $F1_{Macro}$ score.

Q#	SVM	PS	RF	PR	CNN-SVM	1D-CNN	2D-CNN	CNN-LSTM	SEACAT-Net
300	0.3672	0.3427	0.3943	0.3919	0.3263	0.3497	0.4245	0.4377	0.4720
310	0.5845	0.5817	0.6438	0.6232	0.4999	0.6072	0.6557	0.6634	0.6731
401	0.2805	0.3051	0.4132	0.4122	0.2428	0.3760	0.4411	0.4346	0.4510
410	0.3575	0.3666	0.6420	0.6489	0.5030	0.6510	0.6831	0.6701	0.7025
450	0.5584	0.5669	0.5938	0.5742	0.4644	0.5893	0.5969	0.5926	0.6062
453	0.5482	0.5466	0.5945	0.5796	0.4836	0.5734	0.5880	0.6022	0.6156
460	0.2882	0.2877	0.4234	0.4070	0.2981	0.4124	0.4183	0.4137	0.4345
4704	0.6020	0.6054	0.6206	0.6187	0.4945	0.6336	0.6401	0.6636	0.6800
4905	0.4515	0.4617	0.4893	0.4892	0.4910	0.5193	0.5353	0.5312	0.5529
5418	0.2144	0.2704	0.3915	0.3897	0.2705	0.3344	0.3815	0.4162	0.4209

In addition to the $F1_{Macro}$ score, we also employed another measure, *average class accuracy* (ACA), for performance comparison. ACA is calculated as

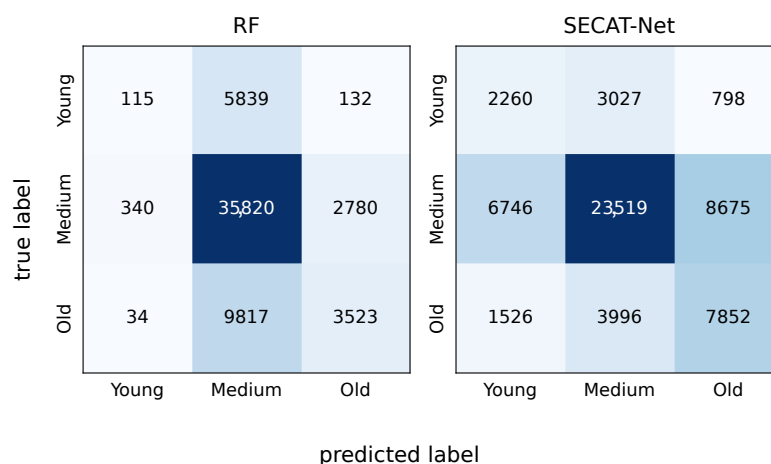
$$ACA = \frac{1}{C} \sum_{i=1}^C Re_i \quad (7)$$

where C is the number of classes and Re_i is the recall value obtained for the i -th class. ACA is the arithmetic mean of per-class accuracy, which is more suitable for imbalanced data than the overall *accuracy* obtained by Equation (2). The results are shown in Table 5. Although ACA is not the measure adopted for architecture evaluation in our search algorithm, SEACAT-Net still outperforms all baselines in eight out of ten questions, while 1D-CNN and CNN-LSTM achieve the best ACA in questions 460 and 5418, respectively. This further demonstrates that automatically designed DNN architectures are more capable of capturing discriminatory information specific to different questions than manually designed architectures. One interesting observation is that RF, as the best-performing machine learning-based baseline method, exhibits significantly worse performance than the proposed method and the majority of the deep learning-based baseline methods in terms of ACA. This indicates that deep learning-based methods handle imbalanced data better than machine learning-based methods.

To further illustrate this, the results for question 300 are used as an example. This particular question is highly imbalanced, with the *Medium* class accounting for 66.7% of the data. Figure 7 displays the confusion matrices obtained by RF and SEACAT-Net. It is evident that RF correctly classifies a significantly higher number of samples from the *Medium* class compared to SEACAT-Net. On the other hand, RF performs considerably worse in classifying the other two minor classes, resulting in a significantly lower ACA value of 0.4007 compared to the ACA value of 0.5271 achieved by SEACAT-Net. This underscores the rationale for selecting $F1_{Macro}$ as the performance measure for the search algorithm, as our goal is to search for an architecture that achieves outstanding balanced classification performance across all classes.

Table 5. Performance comparison in terms of test *ACA*.

Q#	SVM	PS	RF	PR	CNN-SVM	1D-CNN	2D-CNN	CNN-LSTM	SEACAT-Net
300	0.3660	0.3601	0.4007	0.3960	0.3310	0.4578	0.5157	0.5215	0.5271
310	0.5888	0.5866	0.6336	0.6159	0.5008	0.6273	0.6856	0.6866	0.6979
401	0.3805	0.3950	0.4280	0.4184	0.3318	0.4351	0.4612	0.4607	0.4718
410	0.3511	0.3652	0.6311	0.6383	0.5040	0.6731	0.7008	0.6989	0.7064
450	0.5617	0.5669	0.5938	0.5746	0.4984	0.5928	0.6004	0.5978	0.6077
453	0.5544	0.5545	0.5946	0.5798	0.5016	0.5734	0.5880	0.6023	0.6157
460	0.3156	0.2977	0.4221	0.4062	0.3316	0.4498	0.4472	0.4473	0.4378
4704	0.6048	0.6072	0.6143	0.6122	0.4978	0.6401	0.6650	0.6675	0.6790
4905	0.4808	0.4737	0.5186	0.5091	0.4984	0.5212	0.5379	0.5342	0.5531
5418	0.2951	0.3040	0.4018	0.3947	0.3374	0.4027	0.4299	0.4312	0.4249

**Figure 7.** Confusion matrices obtained by RF and SEACAT-Net for question 300.

5. Conclusions

In this paper, we propose an NAS-based method for automatically designing high-performing DNN architectures to address the challenge of identifying customer socio-demographic information using smart meter data. The search space is constructed based on a novel CAFCN architecture, which combines FCN and the channel attention mechanism. We developed a BO-based search algorithm to effectively discover high-performing architectures within this search space. The evaluation measure employed in the search algorithm is the $F1_{Macro}$ score, which is suitable for addressing the imbalanced data commonly encountered in socio-demographic information identification problems.

Performance evaluation and comparison experiments were conducted using a smart meter dataset widely recognized in the research community. The results demonstrate the superiority of our proposed method over machine learning-based approaches and methods employing manually designed DNN architectures. However, identifying customer socio-demographic information solely from smart meter data remains a formidable challenge, as indicated by our results, which show significant room for performance improvement. Our future work involves incorporating more advanced deep learning techniques into the design space and considering seasonal patterns in power consumption data to further enhance identification performance.

Author Contributions: Conceptualization, Z.L. and J.Z.; methodology, Z.L. and J.Z.; software, Z.L. and Q.L.; validation, Z.L., Q.L. and R.Q.; data curation, Z.L., Q.L. and R.Q.; writing—original draft preparation, Z.L., Q.L. and R.Q.; writing—review and editing, Z.L. and J.Z.; visualization, Z.L. and J.Z.; supervision, J.Z.; project administration, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation EPSCoR Cooperative Agreement OIA-1757207, and the Institute for Complex Additive Systems Analysis (ICASA) of New Mexico Institute of Mining and Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Irish CER smart meter dataset is publicly available at <https://www.ucd.ie/issda/data/commissionforenergyregulationcer/> (accessed on 10 November 2024).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Matsumoto, S. How do household characteristics affect appliance usage? Application of conditional demand analysis to Japanese household data. *Energy Policy* **2016**, *94*, 214–223. [CrossRef]
2. Ma, W.; Fang, S.; Liu, G.; Zhou, R. Modeling of district load forecasting for distributed energy system. *Appl. Energy* **2017**, *204*, 181–205. [CrossRef]
3. Lin, J.; Marshall, K.R.; Kabaca, S.; Frades, M.; Ware, D. Energy affordability in practice: Oracle Utilities Opower’s business Intelligence to meet low and moderate income need at Eversource. *Electr. J.* **2020**, *33*, 106687. [CrossRef]
4. Brown, M.A.; Soni, A.; Lapsa, M.V.; Southworth, K.; Cox, M. High energy burden and low-income energy affordability: Conclusions from a literature review. *Prog. Energy* **2020**, *2*, 042003. [CrossRef]
5. Beckel, C.; Sadamori, L.; Santini, S. Automatic socio-economic classification of households using electricity consumption data. In Proceedings of the e-Energy 2013—Proceedings of the 4th ACM International Conference on Future Energy Systems, Berkeley, CA, USA, 21–24 May 2013; pp. 75–86.
6. Beckel, C.; Sadamori, L.; Staake, T.; Santini, S. Revealing household characteristics from smart meter data. *Energy* **2014**, *78*, 397–410. [CrossRef]
7. Hopf, K.; Sodenkamp, M.; Kozlovkiy, I.; Staake, T. Feature extraction and filtering for household classification based on smart electricity meter data. *Comput. Sci. Res. Dev.* **2016**, *31*, 141–148. [CrossRef]
8. Wang, Y.; Chen, Q.; Gan, D.; Yang, J.; Kirschen, D.S.; Kang, C. Deep learning-based socio-demographic information identification from smart meter data. *IEEE Trans. Smart Grid* **2018**, *10*, 2593–2602. [CrossRef]
9. Gajowniczek, K.; Zabkowski, T.; Sodenkamp, M. Revealing Household Characteristics from Electricity Meter Data with Grade Analysis and Machine Learning Algorithms. *Appl. Sci.* **2018**, *8*, 1654. [CrossRef]
10. Fahim, M.; Sillitti, A. Analyzing Load Profiles of Energy Consumption to Infer Household Characteristics Using Smart Meters. *Energies* **2019**, *12*, 773. [CrossRef]
11. Yan, S.; Li, K.; Wang, F.; Ge, X.; Lu, X.; Mi, Z.; Chen, H.; Chang, S. Time–frequency feature combination based household characteristic identification approach using smart meter data. *IEEE Trans. Ind. Appl.* **2020**, *56*, 2251–2262. [CrossRef]
12. Wang, F.; Lu, X.; Chang, X.; Cao, X.; Yan, S.; Li, K.; Duić, N.; Shafie-Khah, M.; Catalão, J.P. Household profile identification for behavioral demand response: A semi-supervised learning approach using smart meter data. *Energy* **2022**, *238*, 121728. [CrossRef]
13. Wang, Y.; Wang, Q.; Wu, Z.; Zhu, B. Socio-demographic Information Extraction from Load Profile Using Convolutional Neural Network. In Proceedings of the 2022 International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID 2022), Xi’an, China, 15–17 April 2022; Atlantis Press: Amsterdam, The Netherlands, 2022; pp. 703–715.
14. Xu, R.; Li, X.D.; Huang, L. A deep learning method for household characteristic classification from smart meter data. In Proceedings of the EMIE 2022; The 2nd International Conference on Electronic Materials and Information Engineering, Hangzhou, China, 15–17 April 2022; pp. 1–5.
15. Fekri, M.N.; Patel, H.; Grolinger, K.; Sharma, V. Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network. *Appl. Energy* **2021**, *282*, 116177. [CrossRef]

16. Luo, Z.; Qi, R.; Li, Q.; Zheng, J.; Shao, S. ABODE-Net: An Attention-based Deep Learning Model for Non-intrusive Building Occupancy Detection Using Smart Meter Data. In Proceedings of the International Conference on Smart Computing and Communication, New York, NY, USA, 18–20 November 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 152–164.
17. He, X.; Zhao, K.; Chu, X. AutoML: A survey of the state-of-the-art. *Knowl. Based Syst.* **2021**, *212*, 106622. [\[CrossRef\]](#)
18. Song, Y.; Wang, A.; Zhao, Y.; Wu, H.; Iwahori, Y. Multi-Scale Spatial–Spectral Attention-Based Neural Architecture Search for Hyperspectral Image Classification. *Electronics* **2023**, *12*, 3641. [\[CrossRef\]](#)
19. Li, Q.; Luo, Z.; Qi, R.; Zheng, J. Automatic Searching of Lightweight and High-Performing CNN Architectures for EEG-Based Driving Fatigue Detection. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–11. [\[CrossRef\]](#)
20. Sharifi, A.A.; Zoljodi, A.; Daneshtalab, M. TrajectoryNAS: A Neural Architecture Search for Trajectory Prediction. *Sensors* **2024**, *24*, 5696. [\[CrossRef\]](#)
21. Karim, F.; Majumdar, S.; Darabi, H.; Harford, S. Multivariate LSTM-FCNs for time series classification. *Neural Netw.* **2019**, *116*, 237–245. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Hao, Y.; Cao, H. A new attention mechanism to classify multivariate time series. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan, 11–17 July 2020.
23. Li, Q.; Luo, Z.; Qi, R.; Zheng, J. DeepTPA-Net: A Deep Triple Attention Network for sEMG-Based Hand Gesture Recognition. *IEEE Access* **2023**, *11*, 96797–96807. [\[CrossRef\]](#)
24. Liu, T.; Liu, Y. GMTP: Enhanced Travel Time Prediction with Graph Attention Network and BERT Integration. *AI* **2024**, *5*, 2926–2944. [\[CrossRef\]](#)
25. Zhang, W.; Chen, G.; Zhuang, P.; Zhao, W.; Zhou, L. CATNet: Cascaded attention transformer network for marine species image classification. *Expert Syst. Appl.* **2024**, *256*, 124932. [\[CrossRef\]](#)
26. Zhang, W.; Li, Z.; Li, G.; Zhou, L.; Zhao, W.; Pan, X. AGANet: Attention-Guided Generative Adversarial Network for Corn Hyperspectral Images Augmentation. *IEEE Trans. Consum. Electron.* **2024**. [\[CrossRef\]](#)
27. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
28. Frazier, P.I. Bayesian optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*; Informs: Hanover, MD, USA, 2018; pp. 255–278.
29. Ma, L.; Cui, J.; Yang, B. Deep neural architecture search with deep graph bayesian optimization. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Thessaloniki, Greece, 14–17 October 2019; pp. 500–507.
30. Zhang, Q.; Hu, W.; Liu, Z.; Tan, J. TBM performance prediction with Bayesian optimization and automated machine learning. *Tunn. Undergr. Space Technol.* **2020**, *103*, 103493. [\[CrossRef\]](#)
31. White, C.; Neiswanger, W.; Savani, Y. Bananas: Bayesian optimization with neural architectures for neural architecture search. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 10293–10301.
32. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. In Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS 2011), Granada, Spain, 12–14 December 2011; Volume 24, pp. 1–9.
33. Schulz, E.; Speekenbrink, M.; Krause, A. A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *J. Math. Psychol.* **2018**, *85*, 1–16. [\[CrossRef\]](#)
34. Hoffman, M.; Brochu, E.; De Freitas, N. Portfolio Allocation for Bayesian Optimization. In Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011), Barcelona, Spain, 14–17 July 2011; pp. 327–336.
35. Commission for Energy Regulation (CER) Smart Metering Project, Irish Social Science Data Archive. 2012. Available online: <https://www.ucd.ie/issda/data/commissionforenergyregulationcer/> (accessed on 10 November 2024).
36. Müller, R.; Kornblith, S.; Hinton, G.E. When does label smoothing help? In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; Volume 32, pp. 1–10.
37. Eunus, S.I.; Hossain, S.; Ridwan, A.; Adnan, A.; Islam, M.S.; Karim, D.Z.; Alam, G.R.; Uddin, J. ECARRNet: An Efficient LSTM-Based Ensembled Deep Neural Network Architecture for Railway Fault Detection. *AI* **2024**, *5*, 482–503. [\[CrossRef\]](#)
38. Chen, Y.; Liu, X.; Rao, M.; Qin, Y.; Wang, Z.; Ji, Y. Explicit speed-integrated LSTM network for non-stationary gearbox vibration representation and fault detection under varying speed conditions. *Reliab. Eng. Syst. Saf.* **2025**, *254*, 110596. [\[CrossRef\]](#)

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.