IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 August 2023, accepted 28 August 2023, date of publication 4 September 2023, date of current version 7 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3311641

RESEARCH ARTICLE

Non-Intrusive Load Classification and Recognition Using Soft-Voting Ensemble Learning Algorithm With Decision Tree, K-Nearest Neighbor Algorithm and Multilayer Perceptron

NIEN-CHE YANG, (Member, IEEE), AND KE-LIN SUNG
Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

Corresponding author: Nien-Che Yang (ncyang@mail.ntust.edu.tw)

This work was supported in part by the National Science and Technology Council (NSTC), Taiwan, under Grant NSTC 111-2622-8-011-014-SB; and in part by the Delta Electronics (DELTA)–National Taiwan University of Science of Technology (NTUST) Joint Research Center.

ABSTRACT Non-intrusive load monitoring (NILM) detects the energy consumption of individual appliances by monitoring the overall electricity usage in a building. By analyzing voltage and current characteristics, NILM can recognize the usage patterns of various appliances, thus facilitating energy conservation and management. To implement non-intrusive load classification and recognition more effectively, this study proposes an ensemble learning algorithm based on soft voting, which comprises a decision tree, K-nearest neighbor algorithm, and multilayer perceptron ($\text{EL-SV}_{\text{DT-KNN-MLP}}$). In this study, the voltage and current features in the plug-load appliance identification dataset (PLAID) and worldwide household and industry transient energy dataset (WHITED) are used as input data. The dataset is examined thoroughly and preprocessed before it is fed into the $\text{EL-SV}_{\text{DT-KNN-MLP}}$. During preprocessing, six different normalization techniques are applied to the data to improve the accuracy and reliability of the machine-learning model, thus rendering the proposed algorithm more adept at classifying and recognizing appliances. The proposed method is validated by comparing it with other machine learning algorithms in terms of accuracy, precision, recall, and F1 score under the six different normalization methods. For the PLAID, the proposed algorithm can achieve high accuracy scores of 99.79%, 98%, 98.11%, 98.36%, 96.42%, and 98.76% under the min–max normalization, MaxAbs scaler, robust scaler, z-score normalization, L1 normalization, and Yeo–Johnson transformation, respectively. Similarly, for the WHITED, the proposed algorithm can achieve high accuracy scores of 99.31%, 98.14%, 98.3%, 98.35%, 97.65%, and 98.02% under the abovementioned normalization methods. The results show that the proposed $\text{EL-SV}_{\text{DT-KNN-MLP}}$ algorithm outperforms the other ten machine learning algorithms examined in this study.

INDEX TERMS Decision tree, ensemble learning, K-nearest neighbors, multilayer perceptron, non-intrusive load monitoring, normalization.

ACRONYMS

NILM	Non-Intrusive Load Monitoring.
$\text{EL-SV}_{\text{DT-KNN-MLP}}$	Proposed Method.
PLAID	Plug-Load Appliance Identification Dataset.
WHITED	Worldwide Household and Industry Transient Energy Dataset.
AC	Air Conditioner.
EDA	Exploratory Data Analysis.
CV	Cross-Validation.
DT	Decision Tree.
KNN	K-Nearest Neighbor.
MLP	Multilayer Perceptron.

The associate editor coordinating the review of this manuscript and approving it for publication was Ikramullah Lali.

EL Ensemble Learning.
QDA Quadratic Discriminant Analysis.
LR Logistic Regression.
SVM Support Vector Machine.
RF Random Forest.

I. INTRODUCTION

Global climate change has become an increasingly critical issue. Owing to the depletion of fossil fuel resources, countries worldwide are transitioning to low-carbon economies. Energy efficiency improvement focusing on energy conservation and carbon reduction has become a central objective in this transition. Public awareness of energy conservation has increased in recent years. In this context, smart grids have become a global trend for reducing carbon dioxide emission and have received significant attention from researchers and government agencies worldwide. Among the different forms of energy, electricity is considered one of the most important, and its effective management is necessary for power conservation. However, the demand for electricity in recent years has increased significantly.

According to [1], over 35% of the global electricity demand in recent years is attributed to building electricity consumption, including residential buildings. A study in [2] shows that Seoul, the capital city of South Korea, has experienced a significant increase in residential building electricity consumption over the past 20 years since 1996. Additionally, in 2020, Ottawa, Canada, experienced an increase of approximately 12% in average daily household electricity consumption, which can be attributed to the global spread of covid-19 [3].

Therefore, energy conservation is a leading global objective. To achieve energy conservation, researchers have focused on implementing improved power utility systems that can effectively reduce energy consumption and cost. The effectiveness of this utility system has been reported in various studies [4], which showed that providing users with detailed information regarding their energy consumption afforded energy savings of up to 15%. Currently, the only information provided to users (other than the price) is the total energy consumed in a building. Using this information, users can implement the necessary actions to achieve energy-saving goals. The requirement of a demand management system based on feedback highlights the relevance of non-intrusive load monitoring (NILM) for buildings [5], [6], [7]; here, the concept of "NILM" was introduced by Hart [7].

Conserving energy and electricity holds immense significance for the public. A comprehensive understanding of household electrical appliance usage is crucial. Through observation, it becomes evident that certain old appliances may consume a substantial amount of electricity. Therefore, an effective approach to promoting energy conservation and management involves considering the replacement of such appliances. This paper proposes an ensemble learning algorithm-based non-intrusive load classification and recognition method.

NILM is an energy analysis technology that uses machine learning and signal processing techniques to detect the energy consumption of each appliance by monitoring the power consumption of an entire building. By analyzing the voltage and current, NILM can recognize the usage of each appliance, as shown in FIGURE 1. NILM has been acknowledged for its potential to promote individual energy-saving by utilizing the electricity consumption of individual appliances and driving applications that support energy-saving concepts. These applications are expected to significantly reduce the carbon footprint associated with electricity consumption [8].

Several examples based on the classification recognition problem in NILM have been presented. In these examples, training data containing information regarding each appliance are used to learn the models for each appliance. The models used in these examples include factorial hidden Markov models [9], [10], [11], artificial neural networks [12], [13], deep learning [14], [15], and optimization algorithms [16], [17]. Nonetheless, these methods exhibit limitations, such as low accuracy or inadequate recognition capabilities.

A comprehensive review of relevant literature revealed that few studies have used ensemble algorithms for non-intrusive load classification and recognition. Therefore, we herein propose an ensemble algorithm based on soft voting for non-intrusive load classification and recognition to obtain more accurate prediction results.

This study makes several significant contributions and offers novel insights. First, a novel ensemble learning algorithm using soft voting is proposed for non-intrusive load classification and recognition. Second, the effectiveness of the proposed method is evaluated using the plug-load appliance identification dataset (PLAID) [18] and worldwide household and industry transient energy dataset (WHITED) [19], respectively. Finally, six normalization techniques are applied to improve the convergence and accuracy of the proposed algorithm. Moreover, the performance of the proposed algorithm is compared with that of other machine-learning algorithms under these six different normalizations. The results demonstrate that the proposed method achieves high efficiency and reliability for non-intrusive load classification and recognition.

The remainder of this paper is organized as follows: Section II introduces the dataset used in this study. Section III outlines the data-preparation process. Section IV presents the proposed method and evaluation indicators used. Section V presents the experimental results of the proposed method. Finally, Section VI concludes the paper with a summary of the main findings.

II. DATASET

Studies have been conducted in North America to obtain information regarding recorded load applications. Consequently, comprehensive high-frequency open-source datasets have become available for non-intrusive load recognition research. In this study, we used PLAID [18] and WHITED [19].

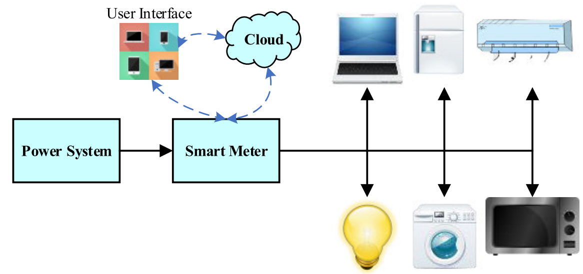

FIGURE 1. Non-intrusive load monitoring (NILM) system.

The PLAID is a publicly available dataset containing labeled voltage and current measurements based on a sampling rate of 30 kHz for various household appliances. The dataset includes the data of 17 different appliance types, with individual measurement records for 330 different brands and models obtained from 65 locations in Pittsburgh, Pennsylvania, including a laboratory setting and 64 households. Each appliance is represented by 5–10 instances, and all the appliances were individually monitored and metered to obtain data pertaining to their actions. Some appliances were monitored while they were being used, and data were obtained within minutes, including information regarding the actions and stoppages of appliances. The features of actions and stoppages were detected as events in the current and voltage signals. FIGURE 2 illustrates a single instance of microwaves in the PLAID.

WHITED is a comprehensive transient energy dataset for residential and industrial sectors. It contains 1259 sets of voltage and current measurements collected from 54 different types of appliances. This publicly available dataset offers valuable insights into energy consumption patterns. The data in WHITED is sampled at a frequency of 44 kHz. FIGURE 3 illustrates a single instance of an air conditioner (AC) in WHITED.

III. DATA PREPARATION

This section presents a detailed description of the data preparation process performed in this study.

A. EXPLORATORY DATA ANALYSIS (EDA)

In exploratory data analysis (EDA), datasets are typically analyzed prior to statistical modeling or machine learning. The purpose of EDA is to provide an initial exploration and understanding of data using various visual and statistical techniques to identify features, associations, and potential problems in a set of data.

Conducting EDA is essential for obtaining deeper understanding into the characteristics and features of the dataset. Various tasks can be performed during EDA, including data profiling, missing-value detection, statistical analysis of data descriptions, and recognition of unique values. The abovementioned processes allow one to obtain precise understanding regarding the dataset characteristics, thus enabling subsequent data preparation steps to be performed effectively. When missing values or outliers are present in a dataset, the appropriate measures must be implemented to ensure the integrity and reliability of the data.

B. APPLIANCE AND ITS FEATURES

In PLAID, information is available on 17 different appliances. Data from ten appliances were specifically chosen for classification and prediction in this study. These ten appliances were selected because they represent common and typical household appliances. Other appliances were excluded due to the limited number of samples available. Therefore, only ten appliances are considered. In WHITED, we focus on appliances with data characteristics similar to those collected in PLAID. For both datasets, the voltage and current data were selected as features for training the machine-learning models. The selected appliances and their corresponding features are listed in TABLE 1.

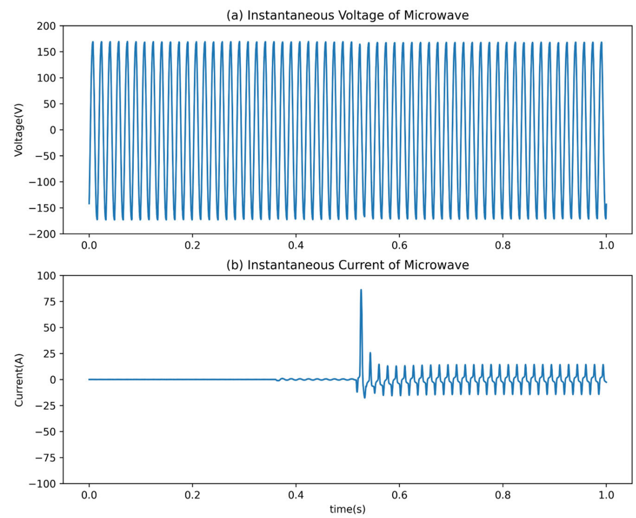

FIGURE 2. One instance for microwave within 1 s.

C. DATA PREPROCESSING

The acquisition of test and validation data is complex, time consuming task, and may not be entirely confidential. Hence, two existing datasets, i.e., the PLAID and the WHITED, were used in this study. However, data preprocessing remains necessary because it can improve data quality, ensure suitability for model training, and enhance the accuracy and reliability of machine learning models.

Label-class separation was performed to separate each appliance type in the dataset. Meanwhile, label encoding was applied to convert categorical data into numerical data, which is necessary for machine-learning models. In the PLAID and WHITED, we selected 60 s and 50 s of measurement data for each appliance as the features, including the voltage and current characteristics, thus resulting in 3.6 million and 4.4 million features per appliance, respectively. A large number of features may increase the time required for training

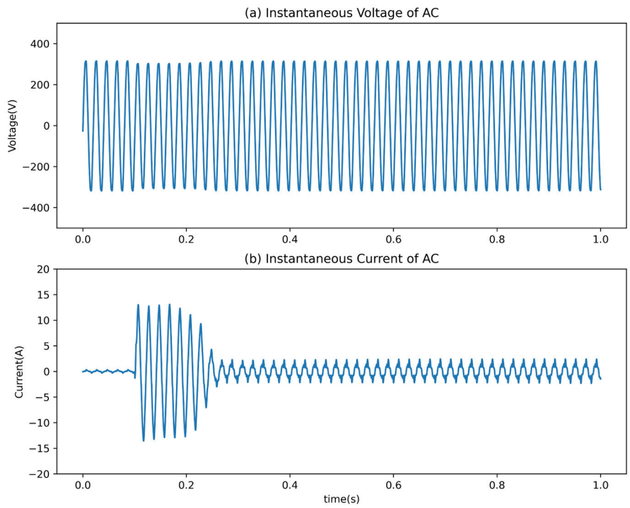

FIGURE 3. One instance for an AC within 1 s.

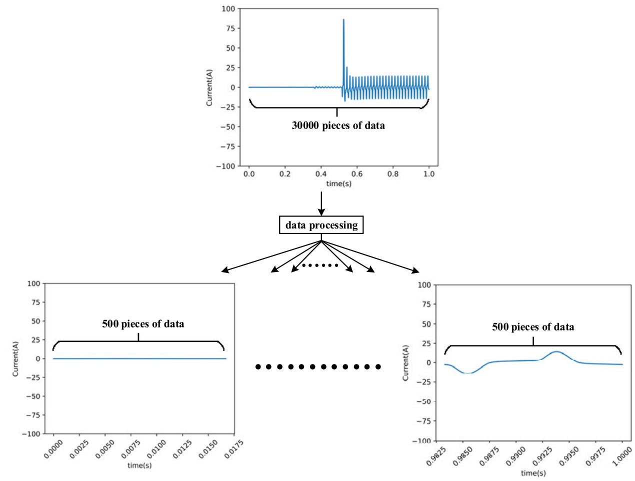

FIGURE 4. Representation of data preprocessing.

machine-learning models. Therefore, feature reduction was performed, in which every 500 sets of data were separated into equal portions. Subsequently, each portion was used as a waveform feature, thereby effectively reducing the number of features, as shown in FIGURE 4. This feature reduction process was also applied in the WHITED. The waveform features were normalized and segregated into 80% training data and 20% test data. TABLE 2 shows the pre-processed results for one appliance feature in each of the two datasets.

TABLE 1. Selected appliances and their features.

Dataset	Appliance	Features
PLAID	Air Conditioner	V, I
	Compact Fluorescent Lamp	V, I
	Fan	V, I
	Fridge	V, I
	Heater	V, I
	Hairdryer	V, I
	Laptop	V, I
	Microwave	V, I
	Vacuum Cleaner	V, I
	Washing Machine	V, I
WHITED	AC	V, I
	Light Bulb	V, I
	Fan	V, I
	Fridge	V, I
	Heater	V, I
	Hair Dryer	V, I
	Laptop	V, I
	Microwave	V, I
	Vacuum Cleaner	V, I
	Washing Machine	V, I

D. NORMALIZATION

Normalization is a crucial preprocessing step that significantly affects classification and was originally termed by Cod in 1972 [20]. Data normalization in machine-learning algorithms offers two advantages: improving model convergence and enhancing accuracy.

Several methods have been used for data normalization, including min–max normalization, MaxAbs scaler, robust scaler, z-score normalization, L1 normalization, and the Yeo–Johnson transformation [21], [22], [23].

TABLE 2. Feature quantity for appliance.

Dataset	Item	Feature	
		Voltage	Current
PLAID	Length of time	60 sec	60 sec
	Original number	1,800,000	1,800,000
	Number after processing	3,600	3,600
	Train set number	2,880	2,880
	Test set number	720	720
WHITED	Length of time	50 sec	50 sec
	Original number	2,200,000	2,200,000
	Number after processing	4,400	4,400
	Train set number	3,520	3,520
	Test set number	880	880

1) MIN–MAX NORMALIZATION

Min–max normalization is a linear transformation method that scales data x into the range [0, 1] and can be calculated using (1).

$$x_{min-max} = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \in [0, 1] \tag{1}$$

where x_i represents an individual data value, x_{max} the maximum value of data x, and x_{min} the minimum value of data x.

Min–max normalization proportionally scales the data from (x_{min}, x_{max}) to the [0, 1] range. Furthermore, min–max normalization preserves the distribution of the original data values and avoids the introduction of bias.

2) MAXABS SCALER

The MaxAbs scaler method rescales data x to fit within the range $[-1, 1]$ and is similar to the previously mentioned min–max normalization method. The MaxAbs scale can be calculated using (2).

$$x_{maxabs} = \frac{x_i}{|x_{max}|}, \in [-1, 1] \tag{2}$$

The MaxAbs scaler method scales data x to the [-1, 1] range while maintaining proportional relationships among features. The advantage of the MaxAbs scaling method is its ability to manage data containing outliers without performing unreasonable scaling. In general, the MaxAbs scaler is a practical feature-scaling method that can improve the accuracy and stability of machine-learning models.

3) ROBUST SCALER

A robust scaler is a data transformation method similar to min–max normalization in terms of the manner by which it transforms data. However, instead of the minimum and maximum values, the median and quartile ranges are used for data scaling to provide a better understanding of the outliers. The robust scaler method is expressed mathematically in (3).

$$x_{robust} = \frac{x_i - x_{mediam}}{Q_3(x) - Q_1(x)} \tag{3}$$

where x_{median} is the median of data x, $Q_1(x)$ the first quartile (25th percentile), and $Q_3(x)$ the third quartile (75th percentile) of the data.

4) Z-SCORE NORMALIZATION

The z-score method, also known as the normalization score method, calculates the ratio of the difference between the target score and mean to the standard deviation. The z-score method accurately reflects the relative standard distance between the target score and mean. If all scores are transformed into z-scores, then the z-score represents the distance from the mean in terms of the standard deviation. By converting the raw scores from a normally distributed dataset into z-scores, the area under the normal curve can be evaluated to assess the percentile rank of raw scores in the original dataset. Z-score normalization can be calculated using (4).

$$x_{z-norm} = \frac{x_i - \mu}{\sigma} \tag{4}$$

where μ is the overall mean and σ is the standard deviation, which is calculated using (5).

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{5}$$

where n is the total number of data points, and $\bar{x}$ is the average set of data x.

5) L1 NORMALIZATION

L1 normalization, also known as the Manhattan distance, is a data-preprocessing technique that scales each sample in a dataset such to achieve the same L1 norm across all samples. The L1 normalization was calculated using (6):

$$x_{L1} = \frac{x_i}{\sum |x_i|} \tag{6}$$

where $\sum |x_i|$ is the sum of the absolute values of each feature in x_i.

L1 normalization allows each feature value of a dataset to be scaled to the same proportion, thereby eliminating the effects of different feature scales. Additionally, it reduces feature redundancy and improves the interpretability and stability of the learning model.

6) YEO–JOHNSON TRANSFORMATION

The Yeo–Johnson transformation is a statistical technique that transforms data into a normal distribution. It involves a formula that includes parameter λ, which can be determined by minimizing the difference between the transformed data and a normal distribution. If data x is non-negative, then it is calculated as shown in (7); if data x is negative, then it is calculated as shown in (8).

$$x_i^{\lambda} = \begin{cases} \left(\dfrac{(x_i+1)^{\lambda} - 1}{\lambda}\right) & if\ \lambda \neq 0, x_i \geq 0 \\ \log(x_i + 1) & if\ \lambda = 0, x_i \geq 0 \end{cases} \tag{7}$$

$$x_i^{\lambda} = \begin{cases} \left(\dfrac{-[(x_i+1)^{(2-\lambda)} - 1]}{2-\lambda}\right) & if\ \lambda \neq 2, x_i < 0 \\ -log(-x_i + 1) & if\ \lambda = 2, x_i < 0 \end{cases} \tag{8}$$

where λ is an arbitrary real number.

The main advantage of using the Yeo–Johnson transformation is its ability to manage cases where the original data contain zero or negative values as well as outliers.

The six aforementioned normalization methods are widely used. Therefore, in this study, these six normalization methods were used for comparison.

E. CROSS-VALIDATION (CV)

Cross-validation (CV) is a technique for evaluating the generalization capability of machine-learning models and is typically used when training data are limited. It involves segregating the dataset into k folds, where one fold is used as the validation set and the remaining k-1 folds are used as the training set. This process is repeated k times to ensure that each fold is used as a validation set at least once. Using this approach, the performance of the proposed model can be effectively evaluated on different datasets, thereby reducing the overfitting of the proposed model to a specific dataset and enhancing the generalization ability of the proposed model.

CV is extensively used in machine-learning algorithms to obtain better nonintrusive load classification and recognition results. In this study, k-fold CV was adopted (with k = 10) and the shuffle data were set as "true." Hence, 10 rounds of testing algorithms were performed based on the aforementioned principles. Each round yielded different results.

IV. PROPOSED METHOD

In this study, a soft-voting ensemble learning method ($\text{EL-SV}_{\text{DT-KNN-MLP}}$) based on a decision tree (DT), the

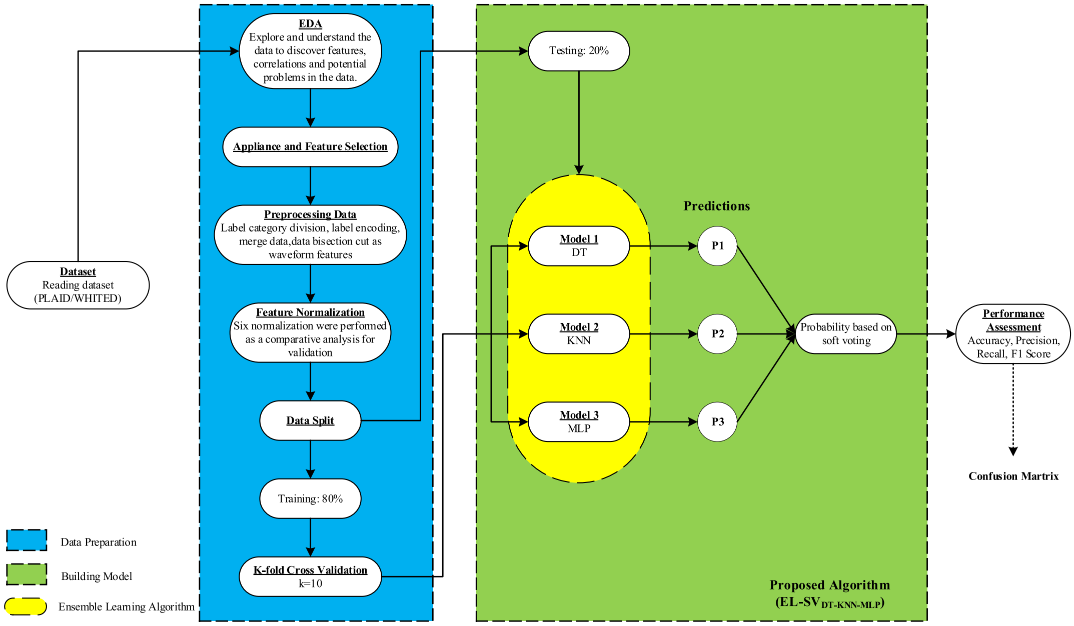

FIGURE 5. Proposed algorithm.

K-nearest neighbor (KNN) algorithm, and multilayer perceptron (MLP) was proposed for non-intrusive load classification and recognition. These three algorithms were selected as the ensemble learning (EL) framework because they complement each other and can improve the accuracy of classification and recognition of features that cannot be distinguished in a single machine-learning algorithm. FIGURE 5 illustrates the process of the proposed method and the learning algorithms used. The soft voting-based EL algorithm allows for the selection of multiple individual machine learning models that can be combined via majority voting or weighted averaging to obtain more stable and accurate prediction results.

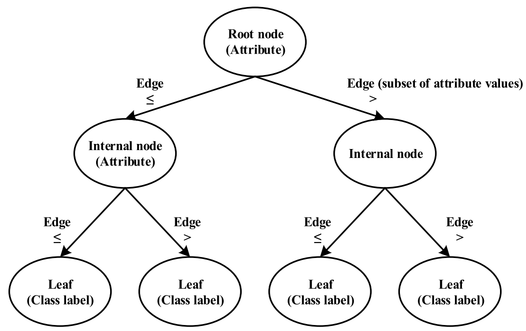

FIGURE 6. Representation of DT.

A. CONVENTIONAL MACHINE-LEARNING MODEL ALGORITHMS

The conventional machine-learning models are artificial intelligence models based on data learning and training. These models employ various steps, such as data acquisition, feature extraction, model selection, and training, to construct a mathematical model that transforms input data into the corresponding output results. The core of conventional machine-learning models is learning patterns and rules from data and using the knowledge obtained to perform tasks, such as predicting and classifying new data. The conventional machine-learning models can be classified into three classes: supervised, unsupervised, and reinforcement. The following section briefly introduces the internal models used in the proposed EL-SV$_{\mathrm{DT-KNN-MLP}}$ algorithm, including the DT, KNN, and MLP.

1) DT

A DT is a supervised learning algorithm based on a tree structure, as shown in FIGURE 6. DTs are typically used to classify and predict data. The main idea of the DT algorithm is to partition the dataset through the nodes of the tree, where each internal node represents a feature or attribute, and each leaf node represents a classification or prediction result. The DT algorithm is executed iteratively, beginning from a root node and then branching out to multiple internal nodes, before ultimately ending at the leaf nodes. During the iteration process, each node retains representative information and removes irrelevant information; as such, the amount of data is gradually reduced until the final iteration [24]. DTs can effectively classify and predict data and have been widely used in practical applications.

When constructing a DT model, specific criteria are used to select the optimal features for splitting. These criteria can be based on entropy or information gain. Entropy is expressed in (9).

$$H(S) = -\sum_{j=1}^{c} P_j \log_2 P_j \tag{9}$$

where c is the total number of classes, and P_j is the probability that class j appears in dataset S.

The information gain is expressed in (10).

$$Gain(S, Z) = H(S) - \sum_{v \in Value(Z)} \frac{|S_v|}{|S|} H(S_v) \tag{10}$$

where Z is the feature to be used in the node, S the total number of results calculated for feature Z, and S_V the number of S subsets.

2) KNN

The KNN algorithm, which was initially proposed by Lemes et al. [25], is a supervised learning algorithm used for classification and regression. It is a nonparametric statistical method that utilizes K training samples closest to the feature space as the input.

In KNN classification, the output is a class label. The classification of an object is determined by the "majority vote" of its KNNs. The KNN algorithm compares the k-nearest points to the input data to identify the class that occurs the most frequently among K points, and then classifies the data accordingly. This process is illustrated in FIGURE 7. The KNN algorithm can be understood and implemented easily; furthermore, it performs exceptionally well on nonlinear datasets for classification.

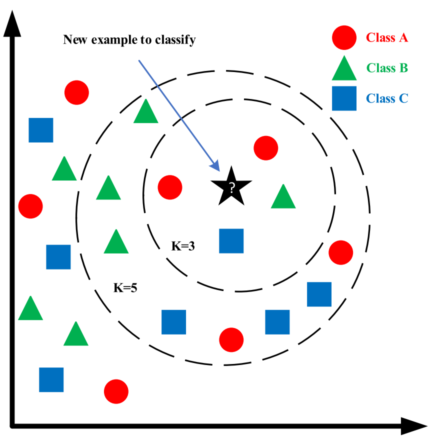

FIGURE 7. Representation of KNN.

The classification steps of the KNN algorithm are as follows:

Step 1. The K value is specified.

Step 2. The distance between the data to be classified and each sample in the training set is calculated. Three typically used methods are used to calculate the distance, as follows:

The Minkowski distance is expressed as

$$D_M = (\sum_{i=1}^{n} |A_i - B_i|^t)^{\frac{1}{t}}, t \in R \tag{11}$$

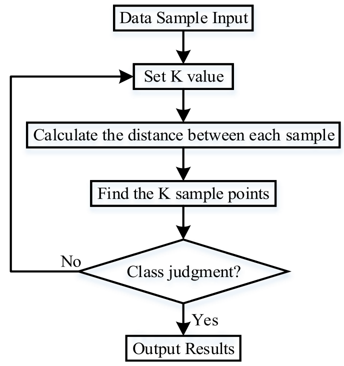

FIGURE 8. Flowchart of KNN classification steps.

The Euclidean distance is expressed as

$$D_E = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2} \tag{12}$$

The Manhattan distance is expressed as

$$D_{Md} = \sum_{i=1}^{n} |A_i - B_i| \tag{13}$$

where D is the calculated distance, A and B are samples in the training set, and n is the total number of data points. In this study, the Manhattan distance calculation was used.

Step 3. After setting the K value, the K-nearest sample points to the data can be recognized, and the group with the highest number of sample points is determined. If the classification is uncertain, then the process is performed again beginning from Step 1 with a different K-value. The classification process is illustrated in FIGURE 8.

3) MLP

The MLP is a feedforward neural network proposed by Rosenblatt in 1950 [26]. It can be regarded as an oriented graph composed of multiple nodes with each layer fully connected to the next layer. Each node in the MLP, except the input nodes, is equipped with a nonlinear activation function that can effectively capture the nonlinear behavior between the input and output vectors.

The backpropagation algorithm is a typically used supervised learning method for training the MLP [27]. The MLP adopts the principles of the human neural system to learn and process data for prediction. After training, the weights are adjusted to reduce errors between the expected and predicted values during training. Mathematically, the output of the MLP is represented as shown in (14) [28]. The basic MLP comprises three layers: an input layer, a hidden layer, and an output layer, as shown in FIGURE 9.

$$y_i = f_E(\sum_{i=0}^{m} w_i x_i) \tag{14}$$

where y_i is the output, m the number of inputs, $f_E()$ the activation function, w_i the weight of the neuron, and x_i the individual data.

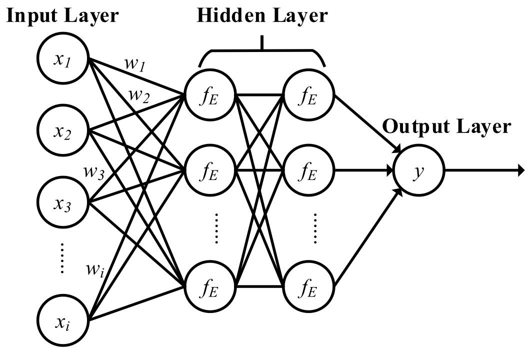

FIGURE 9. Representation of MLP.

B. PROPOSED EL-SV$_{\text{DT-KNN-MLP}}$ ALGORITHM

The EL algorithm is a class of methods that can overcome the limitations of conventional machine-learning models as well as improve their performance [29], [30]. In this study, we propose an EL algorithm based on soft voting, referred to as the EL-SV$_{\text{DT-KNN-MLP}}$ algorithm.

Soft voting is a technique used in EL algorithms to combine the predictions of multiple basic classifiers to obtain a final prediction. Unlike hard voting, which simply counts the majority votes, soft voting considers the prediction confidence or probability of each individual classifier. This approach considers classifiers as a collective group rather than as independent entities. The flowchart of soft voting is shown in FIGURE 10.

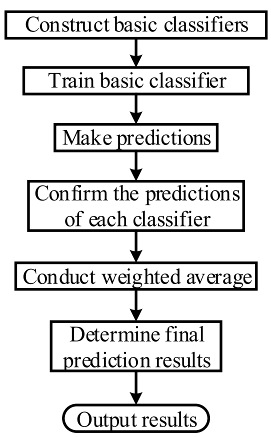

FIGURE 10. Flowchart of soft voting.

In soft voting, classifier predictions are weighted and averaged with probability to obtain a final prediction, as shown in (15).

$$p_{wa}^{j} = \frac{1}{N}\sum_{i=1}^{N} p_i W_i \tag{15}$$

TABLE 3. Tuning hyperparameters for the proposed algorithm.

Model	Items	Parameters
DT	criterion	Entropy
	splitter	Random
	min_samples_split	2
KNN	n_neighbors	10
	p	1
MLP	hidden_layer_sizes	(100, 100)
	Activation	Logistic
	solver	Adam
	random_state	10
	n_iter_no_change	500

where p_{wa}^{j} is the weighted average of the probability of class j, p_i is the predicted probability of classifier i, and W_i is the weighting factor of classifier i. The weighting factors of the three models used in this study are set to 1.

The proposed EL-SV$_{\text{DT-KNN-MLP}}$ algorithm combines three fundamental models: DT, KNN, and MLP. However, these models exhibit certain limitations. DT is prone to overfitting and is sensitive to variations in the data. KNN needs to handle high computational costs, sensitivity to parameter settings, and imbalanced class problems. MLP is susceptible to overfitting and requires good parameter tuning. Various techniques such as EL algorithms, normalization, feature processing, and parameter tuning can be employed to overcome these limitations. Therefore, we chose the EL algorithm based on soft voting to address the challenges in NILM and achieve accurate classification and recognition.

In this study, the GridSearch method was used to tune the hyperparameters of the proposed EL-SV$_{\text{DT-KNN-MLP}}$ algorithm to determine the best parameters for each conventional algorithm. TABLE 3 lists the hyperparameters obtained using the GridSearch method.

The proposed EL-SV$_{\text{DT-KNN-MLP}}$ algorithm for non-intrusive load classification and recognition, as illustrated in FIGURE 5, comprises two main stages, which are discussed below.

- Stage One: Data Preparation
  - Step 1. Perform EDA to acquire an initial understanding of the datasets.
  - Step 2. Select the equipment and features.
  - Step 3. Perform data preprocessing.
  - Step 4. Perform different normalizations.
  - Step 5. Segregate the dataset into training and testing data in a 80/20 ratio.
  - Step 6. Perform CV to train and test the dataset using predefined k-values.
- Stage Two: Model Building
  - Step 1. Define three predefined basic models, namely the DT, KNN, and MLP.
  - Step 2. Use the GridSearch method to optimize the hyper-parameters.
  - Step 3. Train and test the proposed algorithm.
  - Step 4. Determine the final result using the probability based on soft voting.

Step 5. Evaluate the performance based on accuracy, precision, recall, and F1 score for different normalization methods. Present the performance results in a confusion matrix.

C. PERFORMANCE INDICATORS

To evaluate the feasibility of the proposed method for non-intrusive load classification and recognition, we refer to the performance indicators used in [31] and [32], including accuracy, precision, recall, and F1 score.

1) ACCURACY

Accuracy is a widely used performance indicator for non-intrusive load classification and recognition in machine-learning applications. It measures the proportion of correct predictions among all predictions yielded by the proposed method, as expressed in (16).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{16}$$

where TP represents the true positive, TN the true negative, FN the false negative, and FP the false positive.

2) PRECISION

Precision is another typically used performance indicator for non-intrusive load classification and recognition in machine-learning applications. It measures the proportion of TP predictions for all positive predictions yielded by the proposed method, as shown in (17).

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

3) RECALL

The recall indicator measures the proportion of all TP samples correctly predicted as positive values using the proposed method. It is used to evaluate the detection ability of TP samples, as expressed in (18).

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

4) F1 SCORE

The F1 score is a performance indicator for evaluating the effectiveness of machine-learning models. It is the average value of precision and recall; thus, it considers both indicators to balance the model performance in different situations. When the F1 score approaches 1, the proposed method is considered an excellent non-intrusive load classification and recognition model. Conversely, when the F1 score approaches zero, the model performance is regarded as unsatisfactory. The F1 score is expressed in (19).

$$F1Score = \frac{2TP}{2TP + FP + FN} \tag{19}$$

5) CONFUSION MATRIX

A confusion matrix was used to evaluate the performance of the proposed method by classifying the actual observations and model predictions into four classes: TP, TN, FP, and FN. The confusion matrix can evaluate multiple performance indicators such as accuracy, precision, recall, and F1 score. FIGURE 11 shows a confusion matrix for classifying the classes of recognition appliances, where a high-performance method indicates a high TP and TN count, whereas a high FP and FN count indicates an unreliable method.

		Predicted Class	
		1	0
True Class	1	True Positive (TP)	False Positive (FP)
	0	False Negative (FN)	True Negative (TN)

FIGURE 11. Confusion matrix.

V. RESULT AND DISCUSSION

In this study, the proposed EL-SV$_{\mathrm{DT-KNN-MLP}}$ algorithm was tested for the classification and recognition of appliances, and was validated using two datasets, the PLAID and the WHITED, on a computer with ®Core™ i7-11700F@2.50 GHz, 32 GB RAM and ®NVIDIA GEFORCE GTX3060Ti 8G GPU graphics card. The performance of the proposed EL-SV$_{\mathrm{DT-KNN-MLP}}$ algorithm was evaluated using six normalization schemes and compared with those of other machine-learning algorithms. Various evaluation indicators, including accuracy, precision, recall, and F1 score, were used to demonstrate the effectiveness of the proposed non-intrusive load classification and recognition algorithm. Additionally, confusion matrices were used to provide a more intuitive understanding of the performance of the EL-SV$_{\mathrm{DT-KNN-MLP}}$ algorithm.

A. TEST RESULTS

FIGURE 12 shows the 10-fold CV results for the proposed EL-SV$_{\mathrm{DT-KNN-MLP}}$ algorithm. The average accuracy is presented in the form of a bar chart and compared with those of other algorithms, including AdaBoost, DT, KNN, quadratic discriminant analysis (QDA), logistic regression (LR), MLP, support vector machine (SVM), random forest (RF), XGBoost, and Bagging. Among the six normalization methods, the proposed algorithm exhibits the best performance in terms of accuracy for both the PLAID and WHITED datasets.

In PLAID, when applying min-max normalization (as shown in FIGURE 12 (a)), the accuracy of QDA, LR, and SVM is below 90%, whereas the accuracies of AdaBoost, KNN, RF, XGBoost, and Bagging are above 90%. The DT and MLP exhibits an accuracy of approximately 99%. In contrast, the proposed EL-SV$_{\mathrm{DT-KNN-MLP}}$ algorithm achieves a high accuracy of 99.79% for non-intrusive load classification and recognition. Among the remaining five normalization methods, the proposed algorithm achieves high accuracy in appliance classification and recognition.

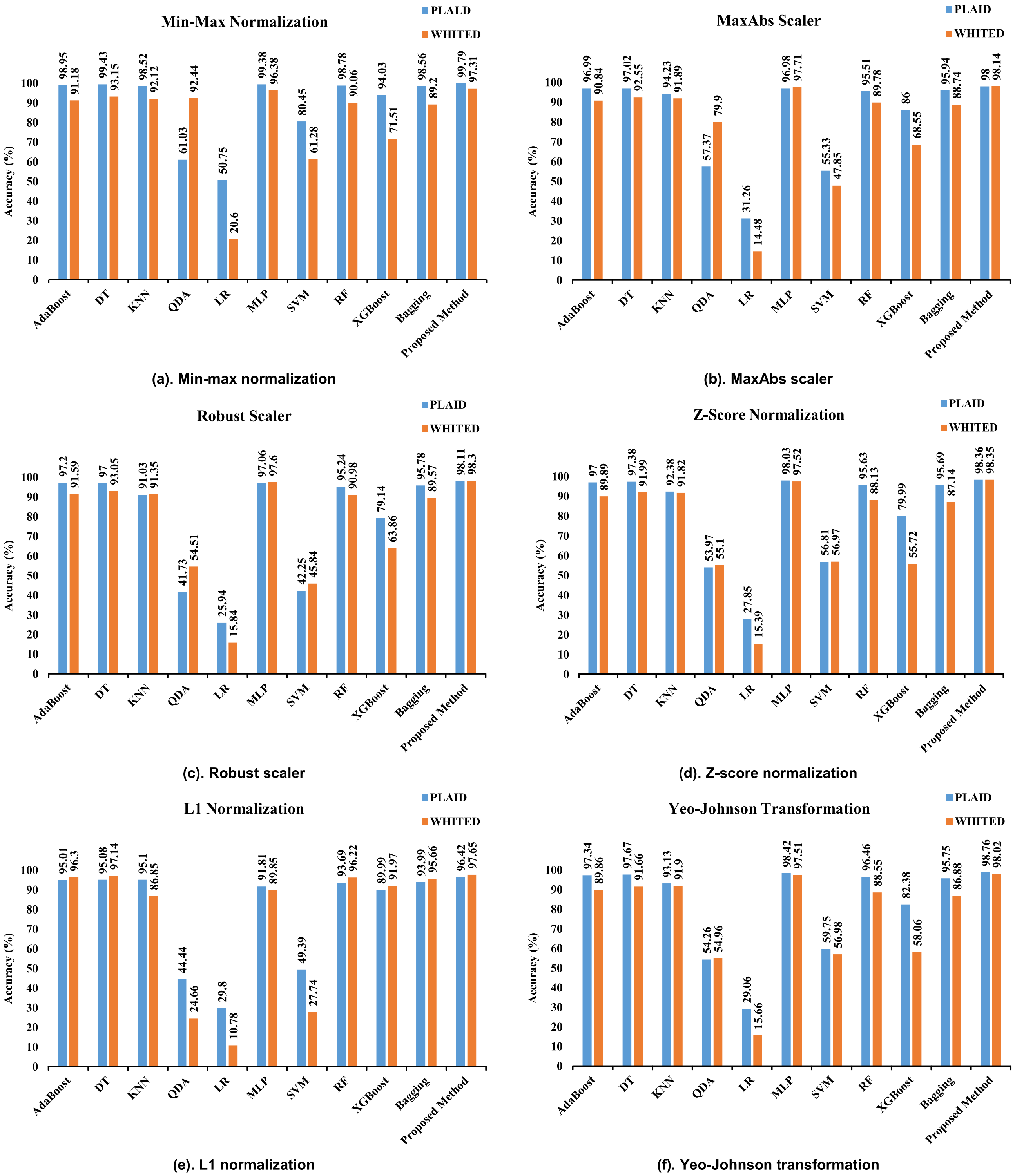

FIGURE 12. **Comparison of accuracy performances.**

When the MaxAbs scaler is used (FIGURE 12 (b)), the proposed algorithm achieves an accuracy of 98%. Similarly, using the robust scaler (FIGURE 12 (c)), the accuracy reaches 98.11%. When z-score normalization is employed (FIGURE 12 (d)), the accuracy is 98.36%. L1 normalization (FIGURE 12 (e)) results in an accuracy of 96.42 %. Finally, applying the Yeo-Johnson transformation (FIGURE 12 (f)) yields an accuracy of 98.76%. In contrast, AdaBoost,

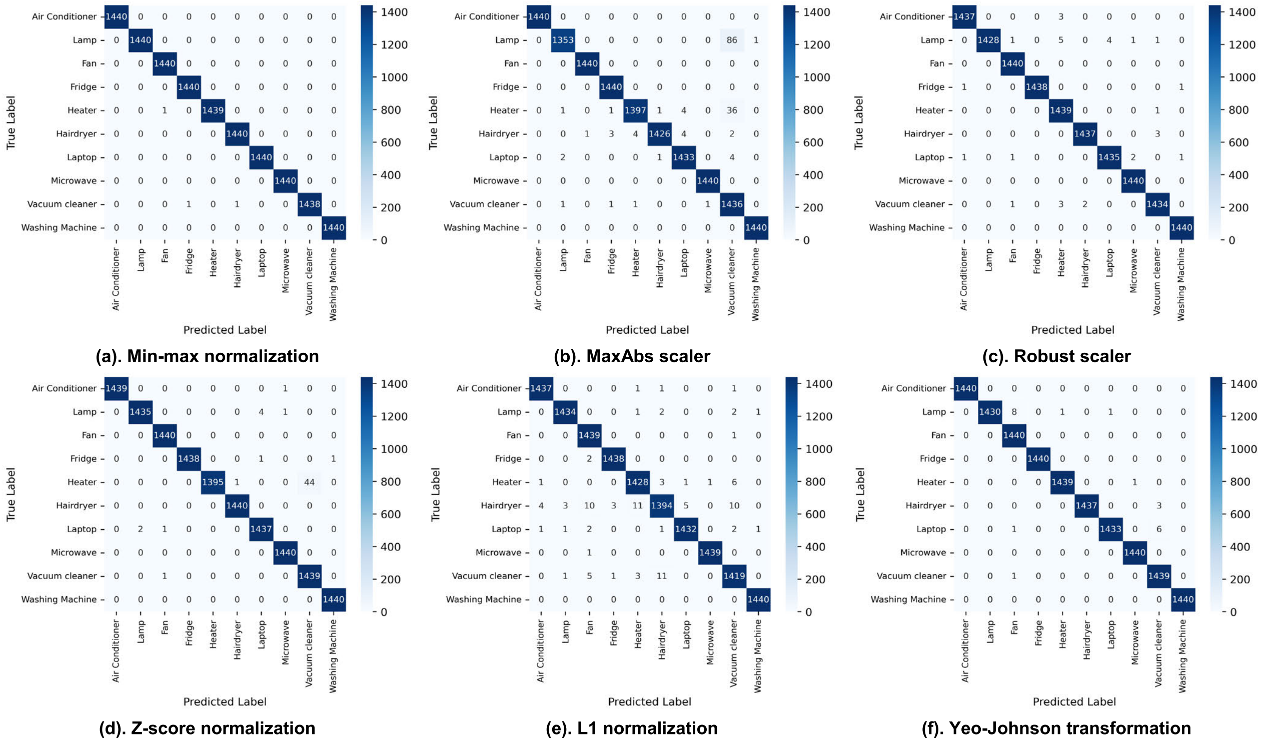

FIGURE 13. Confusion matrix of the proposed algorithm with PLAID.

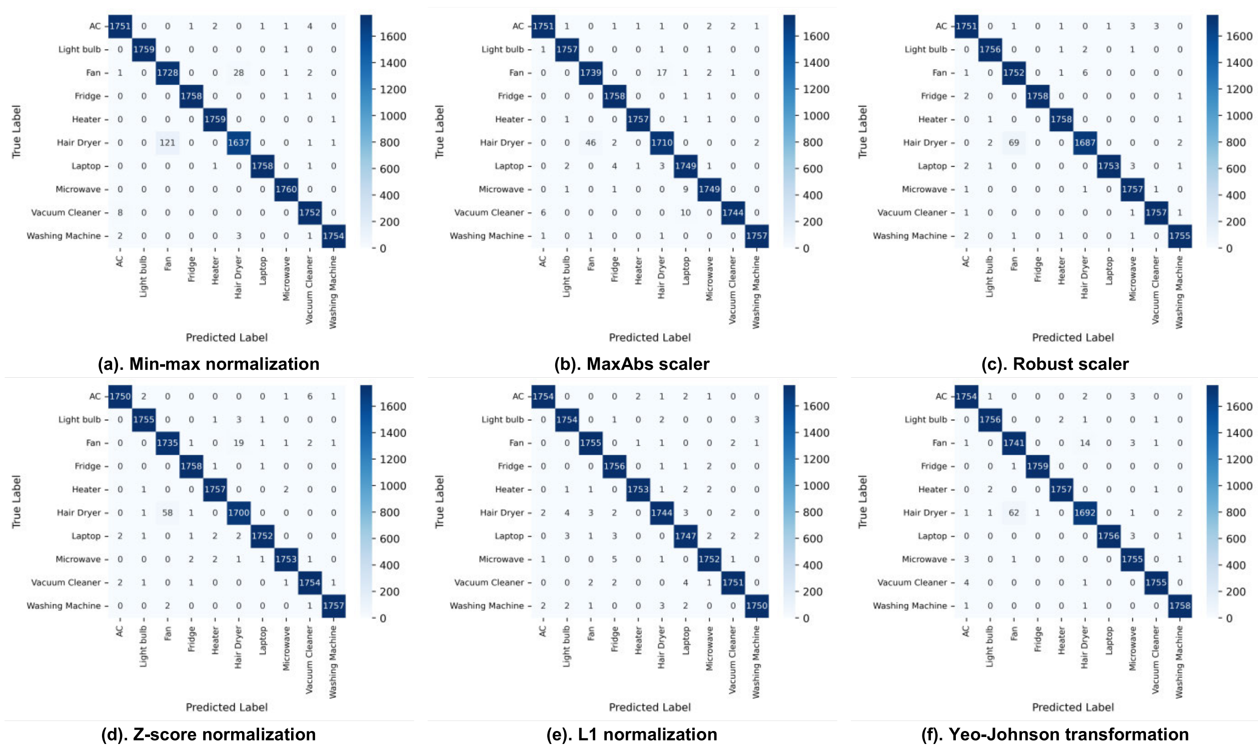

FIGURE 14. Confusion matrix of the proposed algorithm with WHITED.

DT, KNN, MLP, RF, and Bagging exhibit accuracy rates above 90%, whereas those of the other algorithms fall below 90%.

In WHITED, when applying min-max normalization (FIGURE 12 (a)), the accuracy of LR, SVM, XGBoost, and Bagging is below 90%, whereas the accuracies of

TABLE 4. Performance indicators for min–max normalization.

Dataset	Algorithm	Performance Indicators		
		Precision (%)	Recall (%)	F1 Score (%)
PLAID	AdaBoost	98.95	98.95	98.95
	DT	99.43	99.43	99.43
	KNN	98.57	98.52	98.51
	QDA	69.39	61.03	60.49
	LR	52.59	50.75	50.84
	MLP	99.38	99.38	99.38
	SVM	81.01	80.45	80.24
	RF	98.79	98.78	98.78
	XGBoost	94.1	94.03	94.05
	Bagging	98.58	98.56	98.56
	Proposed Method	99.83	99.83	99.83
WHITED	AdaBoost	91.19	91.18	91.18
	DT	93.15	93.15	93.15
	KNN	92.73	92.12	92.03
	QDA	85	82.44	82.04
	LR	17.8	20.6	16.26
	MLP	96.41	96.38	96.38
	SVM	69.84	61.28	60.48
	RF	90.08	90.06	90.07
	XGBoost	73.16	71.51	71.12
	Bagging	89.98	89.2	89.26
	Proposed Method	97.06	97.06	97.06

TABLE 5. Performance indicators for MaxAbs scaler.

Dataset	Algorithm	Performance Indicators		
		Precision (%)	Recall (%)	F1 Score (%)
PLAID	AdaBoost	97	96.99	96.99
	DT	97.03	97.02	97.03
	KNN	94.48	94.23	94.25
	QDA	63.82	57.37	56.12
	LR	32.54	31.26	31.1
	MLP	97.08	96.98	96.98
	SVM	58.43	55.33	55.19
	RF	95.52	95.51	95.51
	XGBoost	86.31	86	86
	Bagging	96.02	95.94	95.94
	Proposed Method	98.05	98.01	98.01
WHITED	AdaBoost	90.84	90.84	90.84
	DT	92.55	92.55	92.54
	KNN	92.5	91.89	91.8
	QDA	82.97	79.9	79.25
	LR	15.05	14.48	14.06
	MLP	97.77	97.71	97.71
	SVM	53.16	47.85	44.08
	RF	89.77	89.78	89.77
	XGBoost	70.25	68.55	67.96
	Bagging	89.47	88.74	88.72
	Proposed Method	97.86	97.86	97.86

TABLE 6. Performance indicators for robust scaler.

Dataset	Algorithm	Performance Indicators		
		Precision (%)	Recall (%)	F1 Score (%)
PLAID	AdaBoost	97.21	97.2	97.2
	DT	97.01	97	97.01
	KNN	92.2	91.03	91.19
	QDA	62.75	41.73	45.25
	LR	32.42	25.94	26.45
	MLP	97.09	97.06	97.06
	SVM	52.2	42.25	43.24
	RF	95.25	95.24	95.24
	XGBoost	79.53	79.14	79.26
	Bagging	95.89	95.78	95.78
	Proposed Method	98.07	98.04	98.05
WHITED	AdaBoost	91.61	91.59	91.6
	DT	93.05	93.05	93.05
	KNN	91.94	91.35	91.24
	QDA	74.53	54.51	56.69
	LR	17.23	15.84	14.73
	MLP	97.67	97.6	97.6
	SVM	67.26	45.84	47.78
	RF	90.99	90.98	90.98
	XGBoost	74.33	63.86	64.31
	Bagging	90.18	89.57	89.56
	Proposed Method	98.18	98.16	98.17

TABLE 7. Performance indicators for Z-score normalization.

Dataset	Algorithm	Performance Indicators		
		Precision (%)	Recall (%)	F1 Score (%)
PLAID	AdaBoost	97.01	97	97
	DT	97.39	97.38	97.39
	KNN	92.8	92.38	92.37
	QDA	64.57	53.97	53.45
	LR	30.74	27.85	28.29
	MLP	98.06	98.02	98.03
	SVM	61.94	56.81	57.32
	RF	95.63	95.63	95.62
	XGBoost	80.32	79.99	79.87
	Bagging	95.8	95.69	95.7
	Proposed Method	98.22	98.17	98.19
WHITED	AdaBoost	89.93	89.89	89.9
	DT	92.01	91.99	92
	KNN	92.39	91.82	91.72
	QDA	73.8	55.1	56.22
	LR	15.67	15.39	14.59
	MLP	97.56	97.52	97.52
	SVM	77.34	56.97	59.78
	RF	88.15	88.13	88.13
	XGBoost	60.97	55.72	55.69
	Bagging	88.02	87.14	87.12
	Proposed Method	97.98	97.95	97.95

AdaBoost, DT, KNN, QDA, MLP, and RF are more than 90%. However, the proposed EL-SV$_{DT-KNN-MLP}$ algorithm achieves the highest accuracy of 97.31% in non-intrusive load classification and recognition. When using the MaxAbs scaler (FIGURE 12 (b)), the proposed algorithm achieves an accuracy of 98.14%. Similarly, using the robust scaler (FIGURE 12 (c)), the accuracy reaches 98.3%. The accuracy of the z-score normalization (FIGURE 12 (d)) is 98.35%. The L1 normalization (FIGURE 12 (e)) results in an accuracy of 97.65 %. Finally, applying the Yeo-Johnson transformation (FIGURE 12 (f)) yields an accuracy of 98.02%. The results show that the proposed algorithm achieves a high accuracy in the classification and recognition of appliances using all six normalization methods.

To better understand the performance of the proposed EL-SV$_{DT-KNN-MLP}$ algorithm, FIGURE 13 and FIGURE 14 the confusion matrices, providing a more intuitive visualization of the classification results. The confusion matrices show the performances of the proposed algorithms for the PLAID and WHITED datasets, considering six different normalization methods.

In the confusion matrix, the diagonal numbers represent the number of features that were predicted correctly. Any numbers in non-diagonal positions indicate incorrect predictions.

TABLE 8. Performance indicators for L1 normalization.

Dataset	Algorithm	Performance Indicators		
		Precision (%)	Recall (%)	F1 Score (%)
PLAID	AdaBoost	95.01	95.01	95.01
	DT	95.1	95.08	95.09
	KNN	95.2	95.11	95.1
	QDA	52.94	44.44	40.34
	LR	33.97	29.8	30.97
	MLP	92.06	91.81	91.83
	SVM	57.27	49.39	50.31
	RF	93.72	93.69	93.7
	XGBoost	90.31	89.99	90
	Bagging	94.4	93.99	94
	Proposed Method	96.35	96.35	96.34
WHITED	AdaBoost	96.31	96.31	96.31
	DT	97.14	97.14	97.14
	KNN	87.93	86.85	86.74
	QDA	21.86	24.66	20.33
	LR	10.69	10.78	10.62
	MLP	90.04	89.85	89.89
	SVM	37.15	27.74	28.59
	RF	96.22	96.22	96.22
	XGBoost	92.09	91.97	91.96
	Bagging	95.77	95.66	95.66
	Proposed Method	97.44	97.44	97.44

TABLE 9. Performance indicators for Yeo-Johnson transformation.

Dataset	Algorithm	Performance Indicators		
		Precision (%)	Recall (%)	F1 Score (%)
PLAID	AdaBoost	97.35	97.34	97.34
	DT	97.68	97.67	97.67
	KNN	93.47	93.13	93.12
	QDA	64.97	54.26	53.76
	LR	32.23	29.06	23.66
	MLP	98.43	98.42	98.42
	SVM	64.83	59.75	60.67
	RF	96.47	96.46	96.46
	XGBoost	82.91	82.39	82.42
	Bagging	95.85	95.75	95.75
	Proposed Method	98.84	98.83	98.83
WHITED	AdaBoost	89.87	89.86	89.86
	DT	91.68	91.66	91.67
	KNN	92.48	91.9	91.8
	QDA	73.57	54.96	56.08
	LR	16.7	15.66	14.9
	MLP	97.58	97.51	97.5
	SVM	77.39	56.98	59.82
	RF	88.55	88.55	88.54
	XGBoost	61.95	58.06	57.24
	Bagging	87.98	86.88	86.89
	Proposed Method	98.11	98.1	98.1

That is, they were incorrectly assigned to other classes. From the PLAID classification results in FIGURE 13, it can be seen that the results are excellent. Nearly every appliance is predicted correctly, with a few mispredictions. The min-max normalization is the most excellent among all the normalization methods, with only three incorrect predictions. From the WHITED classification results in FIGURE 14, it can be seen that the proposed method still performs well in classifying the results. Most of the features are predicted, and only a few features are not predicted accurately. However, there is an issue with the hair dryer appliance classification, where more features were mispredicted and were wrongly labeled as fans. The mispredictions may be attributed to the fact that the hair dryer and fan features are similar after normalization.

TABLE 10. Computation time for each model and proposed method.

Model	DT	KNN	MLP	Proposed Method
Time (sec)	3284.61	3546.87	4892.81	4881.25

TABLE 11. Comparison with related studies.

Reference	Method	Dataset	Accuracy (%)
[32]	BDT	PLAID	92.8
[33]	Stacking EL	PLAID	94.75
		WHITED	97.18
[34]	KNN	PLAID	91.85
		WHITED	92.5
[35]	CNN	PLAID	90.05
		WHITED	91.21
[36]	LSTM	PLAID	92.27
[37]	EL	PLAID	80.7
Proposed Method	EL-$SV_{DT\text{-}KNN\text{-}MLP}$	PLAID	(a).99.79 (Min-Max) (b).98 (MaxAbs Scaler) (c).98.11 (Robust Scaler) (d).98.36 (Z-Score) (e).96.42 (L1) (f).98.76 (Yeo-Johnson)
		WHITED	(a).97.31 (Min-Max) (b).98.14 (MaxAbs Scaler) (c).98.3 (Robust Scaler) (d).98.35 (Z-Score) (e).97.65 (L1) (f).98.02 (Yeo-Johnson)

The precision, recall, and F1 scores of the different algorithms for the six normalization methods in the PLAID and WHITED datasets are tabulated in TABLE 4 to TABLE 9. These three metrics are crucial for evaluating the performance of the algorithms, and achieving values close to 100% indicates optimal results.

The results in TABLE 4 to TABLE 9 indicate that the proposed EL-$SV_{DT-KNN-MLP}$ algorithm outperforms all the other algorithms for all six normalization methods. This demonstrates the superior performance of the proposed method for non-intrusive load classification and recognition.

The computation time is shown in TABLE 10. The proposed method requires less time than MLP, although it is slightly longer than DT and KNN. Although the proposed algorithm may not be the fastest in terms of computation time, it achieves the best classification and identification results. This indicates that the proposed method is effective and useful for practical applications.

B. COMPARISON WITH EXISTING STUDIES

Efforts have been made to compare the proposed study with related studies to demonstrate its reliability in non-intrusive load classification and recognition. TABLE 11 summarizes the comparison of the classification results from the existing load recognition frameworks using two different datasets. The proposed EL-$SV_{DT-KNN-MLP}$ algorithm surpasses the other methods in terms of classification and recognition accuracy rates.

VI. CONCLUSION

In this study, a soft-voting EL algorithm, i.e., EL-SV$_{\mathrm{DT-KNN-MLP}}$, was developed for non-intrusive load classification and recognition. The PLAID and WHITED were used, and an EDA was conducted to verify the data, detect missing values, and describe the data. Voltage and current data were selected as features for classification and recognition. Data preprocessing was performed on selected features, and six different types of normalization were applied. The performance of the proposed EL-SV$_{\mathrm{DT-KNN-MLP}}$ algorithm was benchmarked against and compared with those of other algorithms using various performance indicators under six different types of normalization. The results showed that the proposed algorithm outperformed other algorithms in terms of accuracy, precision, recall, and F1 score.

Moreover, the results obtained from the aforementioned performance metrics demonstrate that the proposed method outperforms other machine learning models for six different normalization methods. This observation confirms that the proposed method consistently achieves excellent results across various normalization techniques, highlighting its wide applicability and effectiveness.

Specifically, when tested on the PLAID, the proposed algorithm achieved high accuracy scores of 99.79%, 98%, 98.11%, 98.36%, 96.42%, and 98.76% under min–max normalization, the MaxAbs scaler, the robust scaler, z-score normalization, L1 normalization, and Yeo–Johnson transformation, respectively. Similarly, when tested on the WHITED, the proposed algorithm achieved high accuracy scores of 99.31%, 98.14%, 98.3%, 98.35%, 97.65%, and 98.02% under the respective normalization methods mentioned above. The test results from both datasets demonstrated the excellent performance of the proposed non-intrusive load classification and recognition algorithm.

REFERENCES

[1] M. Ali, K. Prakash, C. Macana, A. K. Bashir, A. Jolfaei, A. Bokhari, J. J. Klemeš, and H. Pota, "Modeling residential electricity consumption from public demographic data for sustainable cities," *Energies*, vol. 15, no. 6, p. 2163, Mar. 2022.

[2] S. G. Yoo and H.-Á. Myriam, "Predicting residential electricity consumption using neural networks: A case study," *J. Phys., Conf. Ser.*, vol. 1072, Aug. 2018, Art. no. 012005.

[3] A. Abdeen, F. Kharvari, W. O'Brien, and B. Gunay, "The impact of the COVID-19 on households' hourly electricity consumption in Canada," *Energy Buildings*, vol. 250, Nov. 2021, Art. no. 111280.

[4] S. Darby, "The effectiveness of feedback on energy consumption," *A Rev. DEFRA Literature Metering, Billing Direct Displays*, vol. 486, no. 2006, p. 26, 2006.

[5] S. Drenker and A. Kader, "Nonintrusive monitoring of electric loads," *IEEE Comput. Appl. Power*, vol. 12, no. 4, pp. 47–51, 1999.

[6] G. W. Hart, "Residential energy monitoring and computerized surveillance via utility power flows," *IEEE Technol. Soc. Mag.*, vol. 8, no. 2, pp. 12–16, Jun. 1989.

[7] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992.

[8] K. Carrie Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? The case of electricity," *Energy Policy*, vol. 52, pp. 213–234, Jan. 2013.

[9] G.-Y. Lin, S.-C. Lee, J. Y.-J. Hsu, and W.-R. Jih, "Applying power meters for appliance recognition on the electric panel," in *Proc. 5th IEEE Conf. Ind. Electron. Appl.*, Jun. 2010, pp. 2254–2259.

[10] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," in *Proc. PMLR*, 2012, pp. 1472–1482.

[11] M. Zeifman, "Disaggregation of home energy display data using probabilistic approach," *IEEE Trans. Consum. Electron.*, vol. 58, no. 1, pp. 23–31, Feb. 2012.

[12] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O'Hare, "Real-time recognition and profiling of appliances through a single electricity sensor," in *Proc. 7th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw. (SECON)*, Jun. 2010, pp. 1–9.

[13] W. L. Chan, A. So, and L. Lai, "Wavelet feature vectors for neural network based harmonics load recognition," in *Proc. Int. Conf. Adv. Power Syst. Control, Operation Manag.*, 2000, pp. 511–516.

[14] N. V. Linh and P. Arboleya, "Deep learning application to non-intrusive load monitoring," in *Proc. IEEE Milan PowerTech*, Jun. 2019, pp. 1–5.

[15] H. Liu, C. Liu, L. Tian, H. Zhao, and J. Liu, "Non-intrusive load disaggregation based on deep learning and multi-feature fusion," in *Proc. 3rd Int. Conf. Smart Power Internet Energy Syst. (SPIES)*, 2021, pp. 210–215.

[16] M. Baranski and J. Voss, "Genetic algorithm for pattern detection in NIALM systems," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 4, May 2004, pp. 3462–3468.

[17] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load signature study—Part I: Basic concept, structure, and methodology," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 551–560, Apr. 2010.

[18] R. Medico, L. De Baets, J. Gao, S. Giri, E. Kara, T. Dhaene, C. Develder, M. Bergés, and D. Deschrijver, "A voltage and current measurement dataset for plug load appliance identification in households," *Sci. Data*, vol. 7, no. 1, p. 49, Feb. 2020, doi: 10.1038/s41597-020-0389-7.

[19] M. Kahl, A. U. Haq, T. Kriechbaumer, and H.-A. Jacobsen, "Whited—A worldwide household and industry transient energy data set," in *Proc. 3rd Int. Workshop Non-Intrusive Load Monitor.*, 2016, pp. 1–4.

[20] E. F. Codd, "Further normalization of the data base relational model," *Data Base Syst.*, vol. 6, pp. 33–64, Jun. 1972.

[21] K. N. A. Halim, A. S. M. Jaya, and A. F. A. Fadzil, "Data preprocessing algorithm for neural network binary classification model in bank tele-marketing," *Int. J. Innov. Technol. Exploring Eng.*, vol. 9, no. 3, pp. 272–277, Jan. 2020.

[22] L. Friedman and O. V. Komogortsev, "Assessment of the effectiveness of seven biometric feature normalization techniques," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2528–2536, Oct. 2019.

[23] M. A. Siddiqi and W. Pak, "An agile approach to identify single and hybrid normalization for enhancing machine learning-based network intrusion detection," *IEEE Access*, vol. 9, pp. 137494–137513, 2021.

[24] J. R. Quinlan, "Learning decision tree classifiers," *ACM Comput. Surv.*, vol. 28, no. 1, pp. 71–72, Mar. 1996.

[25] D. A. M. Lemes, T. W. Cabral, G. Fraidenraich, L. G. P. Meloni, E. R. De Lima, and F. B. Neto, "Load disaggregation based on time window for HEMS application," *IEEE Access*, vol. 9, pp. 70746–70757, 2021.

[26] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[27] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, and M. Ettaouil, "Multilayer perceptron: Architecture optimization and training," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 1, p. 26, 2016.

[28] M. T. C. Olmedo, M. Paegelow, J.-F. Mas, and F. Escobar, *Geomatic Approaches for Modeling Land Change Scenarios. An Introduction*. Cham, Switzerland: Springer, 2018.

[29] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. Cham, Switzerland: Springer, 2012.

[30] M. Bowles, *Machine Learning with Spark and Python: Essential Techniques for Predictive Analytics*. Hoboken, NJ, USA: Wiley, 2019.

[31] R. V. A. Monteiro, J. C. R. de Santana, R. F. S. Teixeira, A. S. Bretas, R. Aguiar, and C. E. P. Poma, "Non-intrusive load monitoring using artificial intelligence classifiers: Performance analysis of machine learning techniques," *Electr. Power Syst. Res.*, vol. 198, Sep. 2021, Art. no. 107347.

[32] T.-T.-H. Le, H. Kang, and H. Kim, "Household appliance classification using lower odd-numbered harmonics and the bagging decision tree," *IEEE Access*, vol. 8, pp. 55937–55952, 2020.
[33] Y. Li, H. Wang, Z. Yang, J. Yang, and Z. Chen, "Stacking ensemble learning-based load identification considering feature fusion by cyber-physical approach," *IEEE Sensors J.*, vol. 23, no. 6, pp. 5997–6007, Mar. 2023.
[34] Y. Himeur, A. Alsalemi, F. Bensaali, A. Amira, C. Sardianos, I. Varlamis, and G. Dimitrakopoulos, "On the applicability of 2D local binary patterns for identifying electrical appliances in non-intrusive load monitoring," in *Proc. Intell. Syst. Conf. (IntelliSys)*, vol. 3. Cham, Switzerland: Springer, 2021, pp. 188–205.
[35] L. De Baets, J. Ruyssinck, C. Develder, T. Dhaene, and D. Deschrijver, "Appliance classification using VI trajectories and convolutional neural networks," *Energy Buildings*, vol. 158, pp. 32–36, Jan. 2018.
[36] Z. Zhou, Y. Xiang, H. Xu, Z. Yi, D. Shi, and Z. Wang, "A novel transfer learning-based intelligent nonintrusive load-monitoring with limited measurements," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–8, 2021.
[37] H. Liu, H. Wu, and C. Yu, "A hybrid model for appliance classification based on time series features," *Energy Buildings*, vol. 196, pp. 112–123, Aug. 2019.

NIEN-CHE YANG (Member, IEEE) was born in Keelung, Taiwan, in 1977. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2002, 2004, and 2010, respectively. Since 2018, he has been a Faculty Member with the National Taiwan University of Science and Technology, where he is currently an Associate Professor of electrical engineering. His research interests include micro-grid state estimation, harmonic three-phase power flow analysis, probabilistic three-phase power flow analysis, energy loss computation in low voltage networks, micro-grids, smart grids, and electric vehicles. He is a member of the Phi Tau Phi Scholastic Honor Society.

KE-LIN SUNG was born in Taichung, Taiwan, in 1998. He received the B.S. degree from the National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan, in 2021. He is currently pursuing the M.S. degree with the Electrical Engineering Department, National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include machine learning, data analysis, and smart grids.