Statistical Features Extraction and Performance Analysis of Supervised Classifiers for Non-Intrusive Load Monitoring

Md. Mehedi Hasan, Dhiman Chowdhury, *Member, IAENG* and Abu Shahir Md. Khalid Hasan

***Abstract*—An approach to extract distinctive statistical features embedded in current and power signatures of different electrical appliances to substantiate efficacious classification for non-intrusive load monitoring (NILM) is presented in this letter. Supervised classifiers - naïve Bayes, multi-class support vector machine (SVM), ensemble, binary decision tree (DT) and discriminant analysis are employed for performance evaluation based on the extracted feature values. The testbed is COOLL NILM public dataset constituted by 42 devices of different power ratings. The training and testing accuracies along with cross-validation losses associated with each classification algorithm are determined. As a comparative analysis, binary DT classifier produces the best results. Performance assessment corroborates the reliability of the proposed framework for NILM applications.**

***Index Terms*—Current and power signatures, load disaggregation, NILM, statistical features, supervised classifiers**

I. Introduction

NON-intrusive load monitoring (NILM) determines individual energy consumption profile of different electrical appliances of a residential or commercial building without accessing to the individual components. Using a single-point sensor, this technique discerns the individual loads by disaggregating the accumulated energy consumption data on the basis of some methodological approaches. In the age of emerging smart grid technologies, sophisticated home energy management systems and efficacious utility infrastructures, NILM yields to be a crucial tool for reliable and inexpensive smart metering systems.

The concept of non-intrusive appliance load monitoring (NIALM) or NILM was first introduced by George W. Hart [1]. In recent years several novel methodologies have been proposed, which can significantly contribute in regard to non-intrusively load disaggregation [2] - [18]. In this letter, a novel approach to extract inherent statistical features of real-time load signatures is articulated. Then these feature values are applied for classification of different types of loads. Comparative analysis is conducted for classification performance of naïve Bayes, multi-class support vector machine (SVM), ensemble, binary decision tree (DT) and discriminant analysis methods.

Md Mehedi Hasan is with the Department of Electrical and Electronic Engineering, Jessore University of Science and Technology, Jessore 7408, Bangladesh.

Dhiman Chowdhury is with the Department of Electrical Engineering, University of South Carolina, Columbia, SC 29208, USA and (e-mail:dhiman@email.sc.edu).

Abu Shahir Md. Khalid Hasan is with the Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000, Bangladesh.

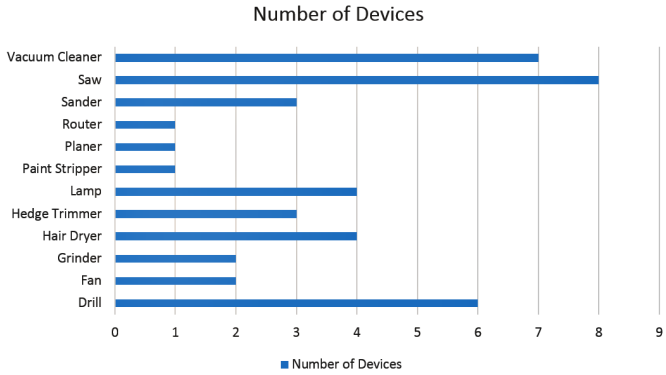

Fig. 1. Device instances present in COOLL NILM public domain dataset

The testbed is the controlled on/off loads library (COOLL) NILM public domain dataset [19]. There are 42 devices of different brands and power ratings in total. Each device has 20 instances of data sets; hence there are 840 datasets in the entire database. Current and voltage data of 6 s with 100k samples per second are present in every set. This work focuses on current and power datasets, where power data are obtained by multiplying corresponding voltage and current data. The devices are of major residential load types. Fig. 1 presents the device types and number of device instances, those constitute the test database.

This work analyzes the current and power signals of the devices and extracts six distinctive statistical features-interquartile range (IQR), crest factor (CF), variance, kurtosis, 6^{th} order moment and mean absolute deviation (MAD). Based on these feature values, a classification model comprising 42 classes is developed and tested in MATLAB ®. Five supervised machine learning classification algorithms are applied for comparative premises of the model performance. 90 % of the database is employed as the training data, whereas 10 % is employed as the testing data for both current and power signatures classification. However, binary DT classifier generates the best classification accuracy for both current and power data among the applied algorithms. For current data, the best classification accuracy obtained from the proposed 6-feature test system is 92.2619 % with the least cross-validation loss of 0.1132. For power data, the best classification accuracy obtained from the proposed 6-feature test system is 86.9048 % with the least cross-validation loss of 0.1825. However, the best training accuracies for both current and power data obtained from the 6-feature system are 97.2039 % and 95.5026 % respectively.

This letter reports extraction of potential statistical feature

TABLE I
EXTRACTED FEATURE COMPONENTS

Index	Features
1	Interquartile Range (IQR)
2	Crest Factor (CF)
3	Variance
4	Kurtosis
5	6^{th} Order Moment
6	Mean Absolute Deviation (MAD)

components from COOLL NILM dataset for both current and power data, which is not reported in the previous NILM works. However, the features are determined by intricate experiments and analysis of the load signatures. This letter presents performance comparisons of different supervised classifiers for NILM applications. The classification model is a multi-class system, which produces very considerable classification accuracies for both types of load signatures. The device classification accuracies obtained here are very comparable with and most cases are superior to those reported in the earlier NILM frameworks.

The remainder of the manuscript is organized as follows. Section II describes the extracted statistical feature components. Section III explains the supervised classifiers, the developed classification model and the performance evaluation of the applied classifiers for both current and power data. Finally, Section IV concludes the letter.

II. STATISTICAL FEATURES EXTRACTION AND ANALYSIS

Statistical analysis is employed to determine potential features from the current and power signatures of the device samples. Table I shows the extracted feature components. The proposed features are described in this section.

Interquartile Range (IQR):
IQR refers to the difference between the value below which lie 25 % of the total data, and that below which lie 75 % of the total data. IQR is a measure of variability, based on dividing a distribution (dataset) into quartiles. Quartiles divide the distribution into four equal parts. The values that divide each part are called the first (Q_1), second (Q_2) and third (Q_3) quartiles. Q_1 is the middle value of the first half of the distribution. Q_2 is the median value and Q_3 is the middle value of the second half of the distribution. IQR is equal to $Q_3 - Q_1$.

Crest Factor (CF):
For a waveform, CF is defined as the ratio of the peak value to the effective value. In case of electrical current signal, CF is measured by the ratio of the instantaneous peak current to the root mean square (rms) current.

Variance:
Variance is a measure of how widely the points in a distribution (dataset) are spread about the mean value. Summation of the squared terms denoting arithmetic differences between each of the data points and the mean value of the distribution is divided by one less than the total number of data points in the distribution to measure variance. For a sample distribution, variance can be mathematically expressed as $\sigma^2 = \frac{(s-\mu)^2}{N-1}$; here σ^2 is the variance, s denotes data points of the distribution, μ is the mean value and N denotes the total number of the data points or the size of the distribution.

Kurtosis:
According to statistical viewpoint, kurtosis is defined as a measure of the combined weight of the tails relative to the rest of a distribution (dataset). Thereby, for a normal distribution kurtosis is found to be zero. The tail heaviness of a sample distribution can be determined by kurtosis which is mathematically expressed as $k = \frac{\sum(s_i-\mu)^4}{N\sigma^4}$; here k is the kurtosis, s_i denotes the i^{th} value of s in the sample distribution, μ is the mean value, N denotes the total number of the data points or the size of the distribution and σ is the standard deviation.

6^{th} Order Moment:
The 6^{th} order moment determines a similar measure like the 2^{nd} order moment (variance) but it focuses more on the outliers (tails) of a sample distribution than the 4^{th} order moment (kurtosis). 6^{th} order moment can be mathematically expressed as $o^6 = \frac{\sum(s_i-\mu)^6}{N\sigma^6}$; here o^6 is the 6^{th} order moment, s_i denotes the i^{th} value of s in the sample distribution, μ is the mean value, N denotes the total number of the data points or the size of the distribution and σ is the standard deviation.

Mean Absolute Deviation (MAD):
The mean absolute deviation (MAD) of a sample distribution (dataset) is the average distance between each value and the mean value. It is mathematically expressed as $m = \frac{\sum|s_i-\mu|}{N}$; here m is the MAD, s_i denotes the i^{th} value of s in the sample dataset, μ is the mean value and N denotes the total number of the data points. MAD measures the variability in a dataset.

The ability of extracted features to discern different device samples can be verified by analyzing the corresponding feature values determined from the respective current and power data. Then scatter plots of the feature values can be generated taking a number of device instances in each plot. Thereby, the quality of the proposed feature components is verified and a good basis for an efficient classification model is ensured. In this work, several scatter feature plots considering different device samples are derived, and a few of those are reported in this article. Figs. 2 - 7 present scatter plots of extracted feature values from current and

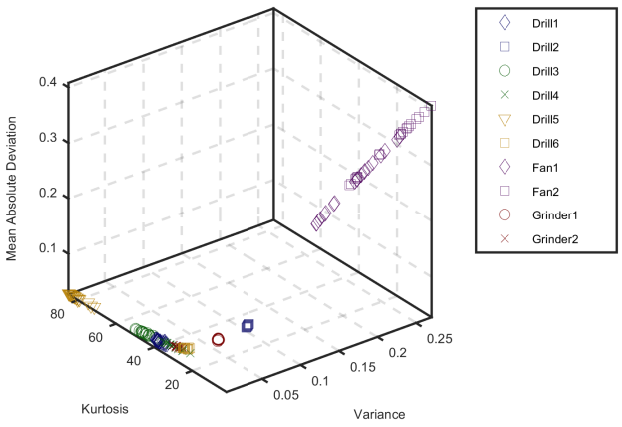

Fig. 2. Feature values extracted from current data of ten device instances (features 6, 4 and 3)

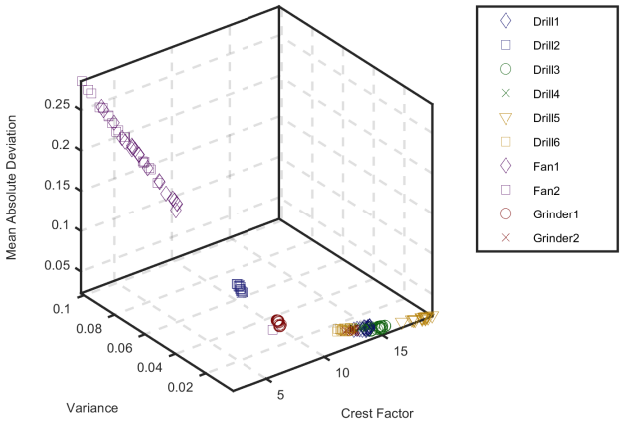

Fig. 3. Feature values extracted from power data of ten device instances (features 6, 3 and 2)

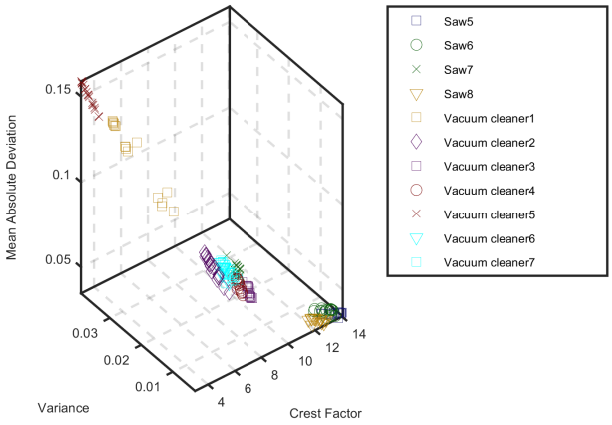

Fig. 5. Feature values extracted from power data of eleven device instances (features 6, 3 and 2)

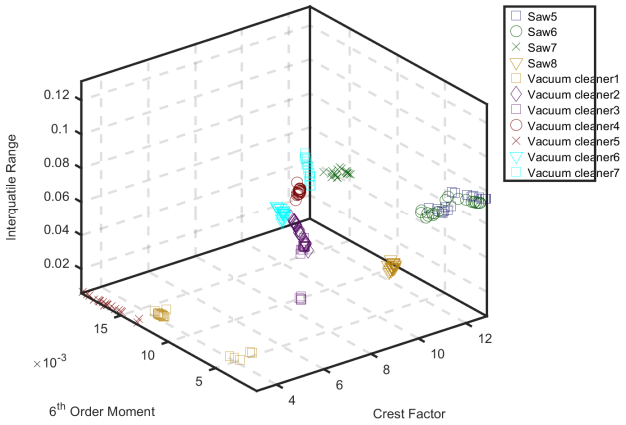

Fig. 4. Feature values extracted from current data of eleven device instances (features 1, 5 and 2)

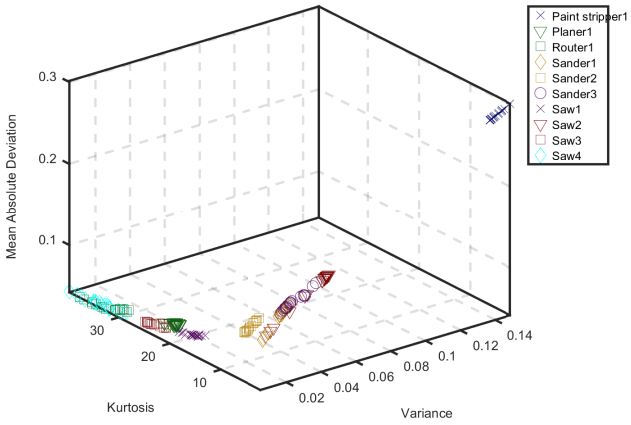

Fig. 6. Feature values extracted from current data of ten device instances (features 6,4 and 3)

power data of a number of device instances. In each of these scatter plots, three features are selected empirically. Figs. 2, 3 and 6 show feature values of the selected candidates for ten device instances. Figs. 4 and 5 show feature values for eleven device samples, whereas Fig. 7 shows feature values for seven device samples. However, the device instances considered for each figure are empirically chosen. Figs. 2, 4 and 6 present scatter feature plots for current data, and Figs. 3, 5 and 7 present scatter feature plots for power data.

III. Classification Model and Performance Analysis

The developed classification model has 42 classes with a training dataset constituting 90 % of the entire database and a testing dataset constituting 10 % of the entire database. This work studies five supervised classification algorithms and evaluates their performance based on the extracted features.

A. Supervised Classifiers

The investigated supervised classification algorithms are briefly explained as follows -

Naïve Bayes:

Naïve Bayes methods are a set of supervised learning algorithms based on applying Bayes theorem with the "naïve" assumption of conditional independence between every pair of features given the value of the class variable [20]. According to [20], for a given class variable Y and dependent

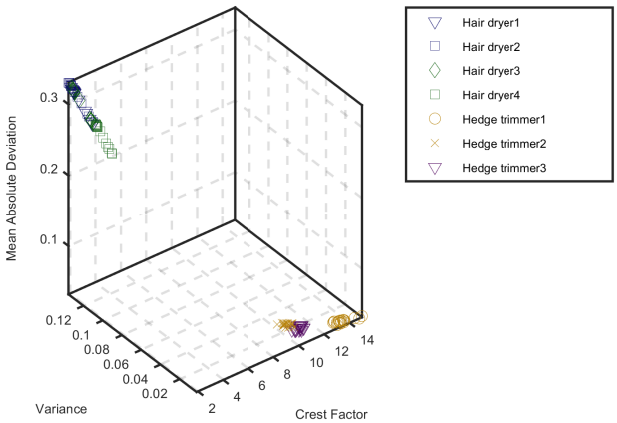

Fig. 7. Feature values extracted from power data of seven device instances (features 6, 3 and 2)

feature vector X_1 through X_p Bayes theorem states that

$$P(Y|X_1,...,X_p) = \frac{P(Y)P(X_1,...,X_p|Y)}{P(X_1,...,X_p)} \quad (1)$$

Applying the conditional independence assumption which implies that

$$P(X_j|Y,X_1,...,X_{j-1},X_{j+1}...,X_p) = P(X_j|Y) \quad (2)$$

For all j, the relationship can be simplified as

$$P(Y|X_1,...,X_p) = \frac{P(Y)\prod_{j=1}^{p} P(X_j|Y)}{P(X_1,...,X_p)} \quad (3)$$

Naïve Bayes classifiers tend to yield posterior distributions that are robust to biased class density estimates, particularly where the posterior is 0.5 (the decision boundary) [21]. Naïve Bayes classifiers assign observations to the most probable class and the algorithm can be explicitly described as follows [21]:

a. Estimates the densities of the predictors within each class.

b. Models posterior probabilities according to Bayes rule. For all $k = 1, ..., K$, it can be mathematically expressed as:

$$\hat{P}(Y=k|X_1,...,X_p) = \frac{\pi(Y=k)\prod_{j=1}^{p} P(X_j|Y=k)}{\sum_{k=1}^{K} \pi(Y=k)\prod_{j=1}^{p} P(X_j|Y=k)} \quad (4)$$

here Y denotes the class index of an observation, $X_1, ..., X_p$ are the random predictors of an observation and $\pi(Y = k)$ represents the prior probability that a class index is k.

c. Classifies an observation by estimating the posterior probability for each class, and then assigns the observation to the class yielding the maximum posterior probability.

If the predictors constitute a multinomial distribution, then the posterior probability $\hat{P}(Y = k|X_1, ..., X_p) \propto \pi(Y = k)P_{mn}(X_1, ..., X_p|Y = k)$, where $P_{mn}(X_1, ..., X_p|Y = k)$ is defined as the probability mass function [21]. Naïve Bayes classifiers require a small amount of training data to estimate the necessary parameters [20]. These classifiers can be extremely fast compared to more sophisticated methods, since the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one dimensional distribution [20]. Thereby the problems associated with dimensionality can be alleviated.

Multi-Class SVM:

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection, which are effective in high dimensional space [20]. SVMs are versatile as regards of different Kernel functions - linear, polynomial, radial basis function (RBF) and sigmoid [20]. However, SVMs do not generate probability estimates directly and if the number of features is much greater than the number of data samples, over-fitting in selecting Kernel functions must be avoided [20]. An SVM constructs a hyper-plane or set of hyper-planes in a high (or infinite) dimensional space in where a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class, called functional margin. Generally the larger the functional margin is, the lower the generalization error of the classifier becomes. According to [20], [22] and [23], the mathematical formulation to implement "one-versus-one" approach for multi-class classification implies that for given training vectors $x_i \in \mathbb{R}^p$, $i = 1, 2, ..., n$ and $y \in [1, -1]^n$, the primal problem can be solved as:

$$\min(\frac{1}{2}\beta^T\beta + C\sum_{i=1}^{n}\zeta_i) \quad (5a)$$

$$\text{subject to, } y_i(\beta^T\phi(x_i) + b) \geq 1 - \zeta_i \quad (5b)$$

$$\text{the dual is: } \min(\frac{1}{2}\alpha^T Q\alpha - e^T\alpha) \quad (5c)$$

$$\text{subject to, } y^T\alpha = 0 \quad (5d)$$

here $\beta \in \mathbb{R}^p$, $C \geq 0$ is the penalty parameter (or regularization parameter), $\zeta_i \geq 0$ denotes slack variables, $\phi(x_i)$ is the hyperplane equation, b is a real number, α_i denotes a parameter implying $0 \leq \alpha_i \leq C$, Q is an $n \times n$ positive semidefinite matrix and e is the vector of all ones. $Q_{ij} = y_i y_j K(x_i, x_j)$, where $K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$ is the Kernel [20], [22], [23]. Training vectors are implicitly mapped into a higher dimensional space by the function ϕ. According to [20], [22] and [23], the decision function is expressed as:

$$\text{sgn}(\sum_{i=1}^{n} y_i\alpha_i K(x_i, x) + \rho) \quad (6)$$

here ρ denotes intercepts. The Kernel chosen in this work is RBF. According to [24], for a class of functions $G(x_1, x_2)$ with a property - $G(x_1, x_2) =< \phi(x_1), \phi(x_2) >$, where ϕ is a function mapping x into a higher dimensional space; RBF Kernel can be defined as:

$$G(x_1, x_2) = \exp(-||x_1 - x_2||^2) \quad (7)$$

Ensemble:

Ensemble methods combine the predictions of several base estimators built with a given learning algorithm in order to improve robustness over a single estimator [20]. There are two families of ensemble methods - averaging methods and boosting methods [20]. The basic principle of averaging methods is to develop several estimators independently and then to average their predictions [20]. In this work an averaging method named random forest (RF) classifier is applied. RF is a classifier consisting of a collection of tree-structured classifiers $[h(x, \theta_k), k = 1, 2, ...]$ where $[\theta_k]$ denotes independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x [25]. In RF algorithm each tree in the ensemble is built from a sample drawn with replacement from the training set [20], [25]. In addition, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features whereas the split that is picked is the best split among a random subset of the features [20], [25]. As a consequence of this randomness, the bias of the forest usually slightly increases but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model, [20], [25].

According to [25], for an ensemble of classifiers $h_1(x), ..., h_K(x)$ and with a training set of random vectors Y, X, the margin function can be defined as:

$$\text{f}(X, Y) = a_k I(h_k(X) = Y) - \max_{j \neq Y} a_k I(h_k(X) = j) \quad (8)$$

TABLE II
TRAINING ACCURACIES (%) OF THE SUPERVISED CLASSIFIERS FOR THE INDIVIDUAL FEATURES FOR CURRENT DATA

Features	Naïve Bayes	Multi-class SVM	Ensemble	Binary DT	Discriminant analysis
IQR	56.3988	44.3690	55.4643	54.5521	47.9167
CF	54.5714	51.7857	56.3988	58.2137	46.6845
Variance	51.4881	40.6190	53.3810	51.5669	46.2798
Kurtosis	58.1845	56.2500	55.3571	53.2417	54.0179
6^{th} order moment	49.8810	39.1702	51.2798	52.1098	47.3512
MAD	53.1250	42.3567	53.2798	55.6667	47.5357

TABLE III
TESTING ACCURACIES (%) AND CROSS-VALIDATION LOSSES OF THE SUPERVISED CLASSIFIERS FOR THE INDIVIDUAL FEATURES FOR CURRENT DATA

Features	Naïve Bayes		Multi-class SVM		Ensemble		Binary DT		Discriminant analysis	
Features	Testing accuracy (%)	Cross-validation loss	Testing accuracy (%)	Cross-validation loss	Testing accuracy (%)	Cross-validation loss	Testing accuracy (%)	Cross-validation loss	Testing accuracy (%)	Cross-validation loss
IQR	50.5952	0.4688	40.6905	0.6667	48.0476	0.5134	50.0134	0.4897	46.6190	0.5327
CF	47.8333	0.4926	42.2619	0.5283	48.6667	0.4998	47.7566	0.4902	44.0476	0.5506
Variance	50.0000	0.5104	33.6130	0.7105	51.8095	0.4993	53.3346	0.4719	46.4286	0.5565
Kurtosis	56.5476	0.4449	53.5714	0.4747	52.5952	0.4896	53.6767	0.4623	51.7857	0.5045
6^{th} order moment	40.5000	0.6810	31.3095	0.8201	44.5238	0.5827	47.5698	0.5567	35.9286	0.6518
MAD	50.1095	0.4697	42.3335	0.6602	49.9322	0.5034	50.2667	0.4898	45.6667	0.5517

here a_k denotes the average number of votes at X, Y and $I(\cdot)$ is the indicator function. The larger the margin is, the more confidence lies in the classification. According to [25], the generalization error is determined as:

$$\delta = P_{X,Y}(f(X,Y) < 0) \tag{9}$$

here $P_{X,Y}$ is the probability over X, Y space. For RF classifiers, $h_k(X) = h_k(X, \theta_k)$ and according to [25], with the increase in tree numbers it becomes converges to

$$P_{X,Y}(P_\theta(h(X,\theta) = Y) - \max_{j \neq Y} P_\theta(h(X,\theta) = j) < 0) \tag{10}$$

Binary Decision Tree:
Decision trees (DTs) are a non-parametric supervised learning method which creates a model that predicts the value of a target variable by learning simple decision rules inferred from the data features [20]. The premier advantages of DTs are: they require little data preparation, they are able to handle both numerical and categorical data, the cost of predicting data is logarithmic in the number of data points used to train the tree model, and they can handle multi-output problems [20]. However, there are a few disadvantages of DTs: they can create over-complex trees that do not generalize the data effectively, they can be unstable due to small variations in the data, and they create biased trees if some classes dominate [20].

Binary DT classifier is capable of multi-class classification on a sample distribution. The detailed mathematical formulation of DT learning method is articulated in [20]. To characterize the classification criteria, if the outcome takes on values $0, 1, ...K-1$, for node m representing a region R_m with N_m observations and given training vectors $x_i \in R_m$ and a label vector y, let us consider

$$P_{mk} = 1/N_m \sum_{x_i \in R_m} I(y_i = k) \tag{11}$$

be the proportion of class k observations in node m. Common measures of impurity are defined as -

Gini:

$$H(X_m) = \sum_k P_{mk}(1 - P_{mk}) \tag{12}$$

Entropy:

$$H(X_m) = -\sum_k P_{mk} log(P_{mk}) \tag{13}$$

and Misclassification:

$$H(X_m) = 1 - max(P_{mk}) \tag{14}$$

where $H(\cdot)$ is an impunity function and X_m is the training data in node m [20].

Discriminant Analysis:
Linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) are two classic methods of classification for multi-class systems having closed form solutions, which can be easily computed [20]. In this work, LDA is applied. According to [20], LDA can be mathematically formulated from probabilistic models, which characterize the class conditional distribution of a sample data $P(X|y = k)$ for each class k. Using Bayes theorem,

$$\begin{aligned} P(y = k|X) &= \frac{P(X|y = k)P(y = k)}{P(X)} \\ &= \frac{P(X|y = k)P(y = k)}{\sum_n P(X|y = n)P(y = n)} \end{aligned} \tag{15}$$

here class k is selected, which maximizes this conditional probability. According to [20], for a more specific application of LDA, $P(X|y)$ is modeled as a multivariate Gaussian distribution with density, which can be expressed as:

$$\begin{aligned} P(X|y = k) = \frac{1}{(2\pi)^{d/2}|\sum_k|^{1/2}} \exp(-\frac{1}{2}(X - \mu_k)^T \\ (\sum_k)^{-1}(X - \mu_k)) \end{aligned} \tag{16}$$

TABLE IV
PERFORMANCE ANALYSIS OF THE SUPERVISED CLASSIFIERS FOR THE PROPOSED 6-FEATURE SYSTEM FOR CURRENT DATA

Classifiers	Training accuracy (%)	Testing accuracy (%)	Cross-validation loss
Naïve Bayes	87.2024	85.1190	0.1607
Multi-class SVM	74.7024	67.2619	0.2961
Ensemble	93.1548	91.0714	0.1205
Binary DT	**97.2039**	**92.2619**	**0.1132**
Discriminant analysis	78.5714	76.7857	0.2649

TABLE V
TRAINING ACCURACIES (%) OF THE SUPERVISED CLASSIFIERS FOR THE INDIVIDUAL FEATURES FOR POWER DATA

Features	Naïve Bayes	Multi-class SVM	Ensemble	Binary DT	Discriminant analysis
IQR	53.3898	44.3346	55.1079	55.2257	45.8847
CF	54.1023	50.6767	55.2286	56.3334	46.4387
Variance	49.5791	43.2556	51.0991	50.8883	45.1198
Kurtosis	52.0667	48.3838	53.6990	54.1212	53.5511
6^{th} order moment	47.3367	41.0334	49.6667	51.6565	42.3890
MAD	50.8889	45.1274	53.5778	54.6559	47.0511

TABLE VI
TESTING ACCURACIES (%) AND CROSS-VALIDATION LOSSES OF THE SUPERVISED CLASSIFIERS FOR THE INDIVIDUAL FEATURES FOR POWER DATA

Features	Naïve Bayes		Multi-class SVM		Ensemble		Binary DT		Discriminant analysis	
Features	Testing accuracy (%)	Cross-validation loss	Testing accuracy (%)	Cross-validation loss	Testing accuracy (%)	Cross-validation loss	Testing accuracy (%)	Cross-validation loss	Testing accuracy (%)	Cross-validation loss
IQR	48.6643	0.4695	41.8732	0.6132	49.0556	0.5077	49.7566	0.4792	44.6313	0.5509
CF	48.1122	0.4899	41.3335	0.5376	50.0375	0.4391	50.2626	0.4388	44.3437	0.5167
Variance	48.9595	0.5202	36.6103	0.6801	50.8532	0.4995	51.1117	0.4533	42.7788	0.5675
Kurtosis	51.6334	0.4650	50.0778	0.4912	51.2432	0.4995	54.4467	0.4370	50.0059	0.4998
6^{th} order moment	39.1667	0.7033	39.2207	0.6988	43.7612	0.6035	47.0509	0.5611	34.9987	0.7266
MAD	49.1250	0.4808	40.6690	0.6931	48.5888	0.5067	50.1094	0.4933	45.5012	0.5549

TABLE VII
PERFORMANCE ANALYSIS OF THE SUPERVISED CLASSIFIERS FOR THE PROPOSED 6-FEATURE SYSTEM FOR POWER DATA

Classifiers	Training accuracy (%)	Testing accuracy (%)	Cross-validation loss
Naïve Bayes	81.7460	76.1905	0.2103
Multi-class SVM	71.2963	67.8571	0.3333
Ensemble	91.6488	80.5714	0.2085
Binary DT	**95.5026**	**86.9048**	**0.1825**
Discriminant analysis	73.6772	72.6109	0.2632

here d is the number of features and μ_k denotes the class means. For LDA, the Gaussians for each class are assumed to share the same covariance matrix: $\sum_k = \sum$ [20].

B. Performance Analysis for Current Data

Table II presents the training accuracy (%) of each supervised classification algorithm obtained for the extracted features for current data. It can be observed that for individual feature performance in case of training accuracy, naïve Bayes, multi-class SVM and discriminant analysis show their most accurate classification performance for kurtosis, whereas ensemble and binary DT show their most accurate performance for CF. However, Table III presents the testing accuracy (%) and cross-validation loss of each of the supervised classification algorithms obtained for the features extracted from current data. From the testing performance analysis, it can be observed that the classifiers show their most accurate performance with the least cross-validation loss for kurtosis.

Table IV presents the performance analysis of the applied supervised classifiers for the proposed 6-feature system for current data. Based on the extracted statistical feature values, the most accurate results are obtained for binary DT algorithm in where the training accuracy is 97.2039 %, the testing accuracy is 92.2619 % and the cross-validation loss is 0.1132. The least accurate results are obtained for multi-class SVM in where the training accuracy is 74.7024 %, the testing accuracy is 67.2619 % and the cross-validation loss is 0.2961. In this work, 10-fold cross-validation losses are measured for each testing case. The performance evaluations are carried out in MATLAB ®.

C. Performance Analysis for Power Data

Table V presents the training accuracy (%) of each supervised classifier obtained for the feature components for

power data. It can be implied that for individual feature performance in case of training accuracy, naïve Bayes, multi-class SVM, ensemble and binary DT show their most accurate classification performance for CF, whereas discriminant analysis shows its most accurate performance for kurtosis. However, Table VI presents the testing accuracy (%) and cross-validation loss of the classification algorithms obtained for each feature derived from power data. From the testing performance analysis, it can be observed that all of the classifiers show their best performance for kurtosis.

Table VII presents the performance analysis of the applied classifiers for the proposed 6-feature test system for power data. Based on the extracted feature components, the most accurate results are obtained for binary DT algorithm in where the training accuracy is 95.5026 %, the testing accuracy is 86.9048 % and the cross-validation loss is 0.1825. The least accurate results are obtained for multi-class SVM in where the training accuracy is 71.2963 %, the testing accuracy is 67.8571 % and the cross-validation loss is 0.3333. Akin to the current data analysis, 10-fold cross-validation losses are measured for testing power data and the performance evaluations are carried out in MATLAB ®.

IV. Conclusion

Non-intrusive load monitoring (NILM) is a significant tool of modern smart grid systems and smart load metering devices. Instead of using multiple sensors for measuring load quantities (voltage, current and power) of multiple electrical appliances, single-point sensor measurement yields more efficient and cost-effective solutions. Therefore, NILM emerges as a very important concept for recognizing individual appliances from the accumulated energy data.

This letter proposes salient and discernible statistical features extraction and performance evaluation of a classification model developed based on the feature values for NILM applications. The classification model comprises five supervised algorithms - naïve Bayes, multi-class support vector machine (SVM), ensemble, binary decision tree (DT) and discriminant analysis, which classify devices present in COOLL NILM public domain dataset to disaggregate loads in an effective way. The current and power signals recorded from the devices are analyzed in this work. The extracted features are - interquartile range, crest factor, variance, kurtosis, 6^{th} order moment and mean absolute deviation, which are applied to develop a 42-class classification model. From a comparative analysis, it is observed that the binary DT classifier performs most accurately with a testing accuracy of more than 92 % for current data and a testing accuracy of more than 86 % for power data. The performance evaluations underscore the efficacy of the presented work for efficient load disaggregation as regards of NILM applications.

The novelty of the presented NILM framework is the statistical features based multi-class classification model for current and power signatures recorded from different electrical devices in COOLL dataset. The statistical features are determined experimentally and comparative analysis of several supervised classification algorithms are reported in this letter, which have not been presented in the earlier NILM works.

References

[1] G. W. Hart, "Nonintrusive Appliance Load Monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870-1891, Dec. 1992.

[2] K. M. Rao, D. Ravichandran and K. Mahesh, "Non-Intrusive Load Monitoring and Analytics for Device Prediction," in Proc. *The International MultiConference of Engineers and Computer Scientists*, pp. 132-136, 2016.

[3] K. Chen, Q. Wang, Z. He, K. Chen, J. Hu and J. He, "Convolutional sequence to sequence nonintrusive load monitoring," *The Journal of Engineering*, vol. 2018, no. 17, pp. 1860-1864, Nov. 2018.

[4] D. Lee, "Phase noise as power characteristic of individual appliance for non-intrusive load monitoring," *Electronics Letters*, vol. 54, no. 16, pp. 993-995, Aug. 2018.

[5] Y. F. Wong, T. Drummond and Y. A. Sekercioglu, "Real-time load disaggregation algorithm using particle-based distribution truncation with state occupancy model," *Electronics Letters*, vol. 50, no. 9, pp. 697-699, May 2014.

[6] B. Zhao, L. Stankovic and V. Stankovic, "On a Training-Less Solution for Non-Intrusive Appliance Load Monitoring Using Graph Signal Processing," *IEEE Access*, vol. 4, pp. 1784-1799, Apr. 2016.

[7] J. M. Gillis and W. G. Morsi, "Non-Intrusive Load Monitoring Using Semi-Supervised Machine Learning and Wavelet Design," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2648-2655, Nov. 2017.

[8] D. Chowdhury and M. M. Hasan, "Non-Intrusive Load Monitoring Using Ensemble Empirical Mode Decomposition and Random Forest Classifier," *The International Conference on Digital Image and Signal Processing (DISP)*, pp. 1, Apr. 2019.

[9] Y.-H. Lin and M.-S. Tsai, "Non-Intrusive Load Monitoring by Novel Neuro-Fuzzy Classification Considering Uncertainties," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2376-2384, Sep. 2014.

[10] H.-H. Chang, K.-L. Lian, Y.-C. Su and W.-J. Lee, "Power-Spectrum-Based Wavelet Transform for Nonintrusive Demand Monitoring and Load Identification," *IEEE Transactions on Industry Applications*, vol. 50, no. 3, pp. 2081-2089, May-Jun. 2014.

[11] M. Gaur and A. Majumdar, "Disaggregating Transform Learning for Non-Intrusive Load Monitoring," *IEEE Access*, vol. 6, pp. 46256-46265, Aug. 2018.

[12] X. Wu, X. Han, L. Liu and B. Qi, "A Load Identification Algorithm of Frequency Domain Filtering Under Current Underdetermined Separation," *IEEE Access*, vol. 6, pp. 37094-37107, Jun. 2018.

[13] S. Singh and A. Majumdar, "Deep Sparse Coding for Non-Intrusive Load Monitoring," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4669-4678, Sep. 2018.

[14] A. Rahimpour, H. Qi, D. Fugate and T. Kuruganti, "Non-Intrusive Energy Disaggregation Using Non-Negative Matrix Factorization With Sum-to-k Constraint," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4430-4441, Nov. 2017.

[15] W. Kong, Z. Y. Dong, J. Ma, D. J. Hill, J. Zhao and F. Luo, "An Extensible Approach for Non-Intrusive Load Disaggregation With Smart Meter Data," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3362-3372, Jul. 2018.

[16] S. M. Tabatabaei, S. Dick and W. Xu, "Toward Non-Intrusive Load Monitoring via Multi-Label Classification," *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 26-40, Jan. 2017.

[17] S. Welikala, C. Dinesh, M. P. B. Ekanayake, R. I. Godaliyadda and J. Ekanayake, "Incorporating Appliance Usage Patterns for Non-Intrusive Load Monitoring and Load Forecasting," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 448-461, Jan. 2019.

[18] D. F. Teshome, T. D. Huang and K.-L. Lian, "Distinctive Load Feature Extraction Based on Fryze's Time-Domain Power Theory," *IEEE Power and Energy Technology Systems Journal*, vol. 3, no. 2, pp. 60-70, Jun. 2016.

[19] T. Picon, M. N. Meziane, P. Ravier, G. Lamarque, C. Novello, J.-C. L. Bunetel and Y. Raingeaud, "COOLL: Controlled On/Off Loads Library, a Public Dataset of High-Sampled Electrical Signals for Appliance Identification," *arXiv preprint arXiv:1611.05803 [cs.OH]*, Nov. 2016.

[20] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[21] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. NY: Springer, 2008.

[22] I. Guyon, B. Boser and V. Vapnik, "Automatic Capacity Tuning of Very Large VC-dimension Classifiers," in *Advances in Neural Information Processing Systems*, Burlington: Morgan Kaufmann, 1993, pp. 147-155.

[23] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, Sep. 1995.

[24] N. Christianini and J Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge, UK: Cambridge University Press, 2000.

[25] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, Oct. 2001.