cigre Turkey

TEPES
TURKISH JOURNAL OF ELECTRICAL POWER AND ENERGY SYSTEMS

TEPES, Vol. 5, Issue. 1, 19-29, 2025 **DOI:** 10.5152/tepes.2024.24030

RESEARCH ARTICLE

Deep Neural Network Implementation for Appliances Identification by Using Current and Voltage Signatures

Yılmaz Güven

Department of Electronics and Automation, Kırklareli University Vocational Technical School, Kırklareli, Türkiye

Cite this article as: Y. Güven, "Deep neural network implementation for appliances identification by using current and voltage signatures," *Turk J Electr Power Energy Syst.*, 2025; 5(1), 19-29.

ABSTRACT

The PLAID database was published earlier last year and contains many households' devices and appliances with numerous high-frequency measurements at different locations. In this paper, the researchers extracted features such as peak to peak, peak to root mean square, minimum, maximum, mean, median value, standard deviation, phase angle, active and reactive power from these current and voltage signals. A multi-layered and SoftMax back-propagated artificial deep neural network (DNN) has been trained and tested with these data. Batch normalization has been used to optimize the DNN. Different architectures, activation functions, and training algorithms have been tried out to get the best results. Then this method was implemented within a low-cost embedded system to identify appliances by using their current and voltage signature. This device provides an identification method using only one sensor within an embedded system, and accuracy of the DNN is slightly better than studies which use the same dataset. On the other hand, deploying trained neural networks on an embedded system can be tricky and overwhelming. This paper also demonstrated that using open standards for machine learning makes these processes and gives interoperability.

Index Terms—Deep learning, embedded system, intelligent measurement, smart grid, smart home

I. INTRODUCTION

Nowadays, everything is about artificial intelligence because of the accumulated data in recent years. Almost every device used in daily life, such as laptops, smart TVs, cars and household appliances, are getting smarter every day. Smart grids and smart buildings are crucial because of the indispensability of electrical energy. In the near future, electric vehicles are also expected to become part of our smart homes. Therefore, smart metering and monitoring will be required for every energy-consuming device.

Nevertheless, an uncomplicated and effective method should detect these devices over powerlines without extra equipment. For this reason, researchers have focused on this identification problem using only the voltage and current signatures of the device. Developments in artificial intelligence technologies have increased the accuracy of identification problems. Machine learning (ML) techniques such as artificial neural networks (ANNs) have achieved good results with the help of some optimization and data pre-processing techniques.

This study has focused on creating an effective deep neural network (DNN) to identify a home appliance using only voltage and current measurement data. Although there are many datasets related to this subject, we have used the Plug-Load Appliance Identification Dataset (PLAID) [1] for this study because of its high frequency measurement and variety of appliances and measurement locations. Table 1 shows other populer appliance datasets such as Controlled on/off Loads Library (COOLL) [2], Appliance Consumption Signatures (ACS-F2) [3], United Kingdom Domestic Appliance-Level Electricity (UK-DALE) [4], Retrofit Electrical Load Measurements (REFIT) [5], and The Reference Energy Disaggregation Data (REDD) [6].

Previous studies have shown that recognition and identification problems can be solved using ML and ANNs. A non-intrusive monitoring methodology [7] with a feed-forward neural network has reached an 83.88% correct classification rate by using the ACS-F2 database. Real-time recognition through a single sensor [8] has achieved an 84% accuracy rate with a shallow ANN. In [9], a Hidden Markov Model was used for appliance and state recognition with

Corresponding author: Yılmaz Güven ylmzguven@hotmail.com

Received: September 26, 2024
Revision Requested: October 6, 2024
Last Revision Received: October 9, 2024
Accepted: October 15, 2024
Publication Date: November 18, 2024

TABLE I.
SPECIFICATION OF VARIOUS DATABASE

Database	Specifications
PLAID[1]	17 different appliances with voltage and current measurements from 65 different locations such as offices, homes, and schools at 30 kHz sampling rate
COOLL[2]	12 different appliances with voltage and current measurement in a laboratory environment at 100 kHz sampling rate
ACS-F2[3]	15 appliances with power parameters in laboratory environment at .1 Hz sampling rate
UK-DALE[4]	7 applications with consumption data from 5 houses at 16 kHz sampling rate
REFIT[5]	18 applications with power parameters from 20 Houses at 10 kHz sampling rate
REDD[6]	8 appliances with power and frequency measurement from 6 houses at a 15 kHz sampling rate

accuracy rates between 90% and 94%. In a study on Appliance identification for different electrical signatures [10] has achieved relatively better accuracy by using some ML algorithms and the moving average method.

All these studies cover conventional ML methods, and they need to be progressed to the next level. Deep learning (DL) is a new, powerful approach to artificial intelligence technologies. While classical ANNs have fully connected nodes with few layers, a DNN can have many connected or independent nodes and layers. Multilayer perceptron, deep belief network (DBN), DNN, convolutional neural network (CNN), recurrent neural network (RNN), and time-delay neural network are some of the DL models [11].

Signal signature and pattern recognition are being used in many different areas. Automatic arrhythmia diagnosis with Electrocardiogram (ECG) signals [12], real-time detection of signal patterns via multichannel spatiotemporal sensor array [13], power quality classification [14], frequency signature tracking [15], fault classification in power electronic circuits [16], and fault analysis in transmission lines [17] are some of the many signal signature and pattern recognition.

This study has focused on building an effective new method for appliance identification using voltage and current signal signatures. The extracted features from these signals have been normalized with Batch Normalization (BN) to optimize the DNN. SoftMax back-propagation with cross-entropy has been used to get the best training results.

The voltage and current signals have been separated into train and test data with a 50% ratio to get minimum validation error.

A brief comparison of performance has also been presented in the results section with the real-time implementation of the method within a low-cost embedded system. There are some prototype systems related to this study such as smart energy management with forecasting and load strategy for renewable systems [18], consumer load measurement for automated buildings [19], and low voltage smart meters for power quality disturbance [20].

Main Points

- Intelligent measurement system for smart grids.
- Low-cost embedded artificial intelligence device.
- Appliance recognition by using voltage and current signatures: deep neural network implementation.

This study demonstrates that DNNs can achieve high performance in low-cost embedded systems using specialized techniques. Using open exchange standards to deploy trained networks makes it better for hardware and software compatibility.

II. METHODOLOGY

DL is a rapidly developing method for solving many different problems. Soft sensor data can be compared to detect hazardous gases using a DBN [21]. Recurrent neural networks can detect acoustic anomalies without a clean sample [22]. Besides, DL methods are being started to be used for electrical load monitoring and planning for Nonintrusive load monitoring (NILM) applications [23]. Convolutional neural networks were first designed for feature extraction and image classification [24], then these methods have been applied to many different problems such as detection of pointers in analog meter display [25] and intelligent fault diagnosis [26].

The PLAID dataset has 1876 individual measurements of 16 different appliances such as air conditioners, coffee maker, refrigerator, heater, vacuum cleaners, washing machine, etc. These measurements have been sampled at 30 kHz. Root mean square (RMS) voltage values vary between 110V and 130V because of voltage differences in the United States of America (USA).

The current values also vary between 0 and 20A depending on appliance type. These devices are labeled with identifying numbers between 1-16. Some appliances have nonlinear load characteristics while others have inductive or resistive load characteristics. On the other hand, physically different types of appliances from different brands have been used for measurement. The summary of the appliances can be seen in Table II.

A. Feature Extraction

The voltage and current signatures have been recorded at high frequency for better feature extraction. The difference in current signatures is obvious, as it can be seen in Fig. 1. However, voltage

TABLE II.
SUMMARY OF THE APPLIANCES IN PLAID

ID	Appliance	Load	Number of Measurement
1	Air conditioner	Nonlinear	204
2	Blender	Inductive	2
3	Coffeemaker	Resistive	10
4	Compact fluorescent	Nonlinear	230
5	Fan	Inductive	220
6	Refrigerator	Inductive	108
7	Hairdryer	Resistive	246
8	Hair iron	Nonlinear	10
9	Heater	Resistive	85
10	Incandescent light	Resistive	157
11	Laptop	Nonlinear	216
12	Microwave oven	Nonlinear	200
13	Soldering iron	Nonlinear	20
14	Vacuum cleaner	Inductive	83
15	Washing machine	Nonlinear	75
16	Water kettle	Resistive	10

signatures of appliances look the same to the naked eye as can be seen in Fig. 2. Therefore, we have extracted some distinguishing features such as peak to peak, peak to RMS (1), minimum and maximum values, mean (2), median, and standard deviation (3) [27]. All extracted features can be seen in Table III.

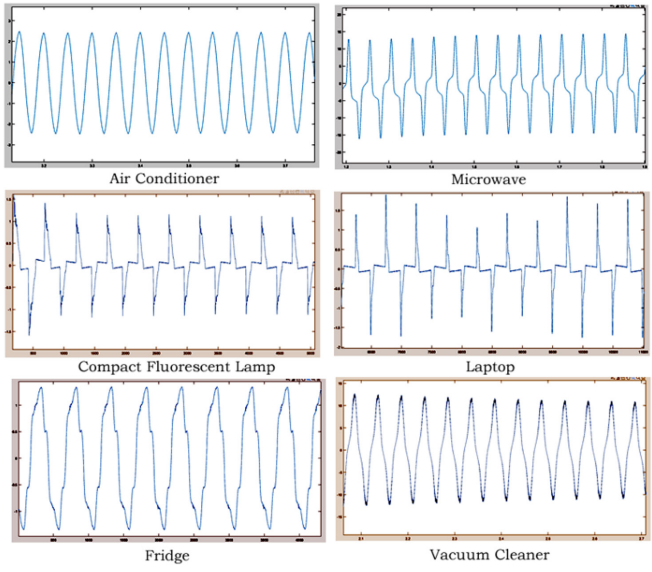

Fig. 1. Current signatures of different appliances.

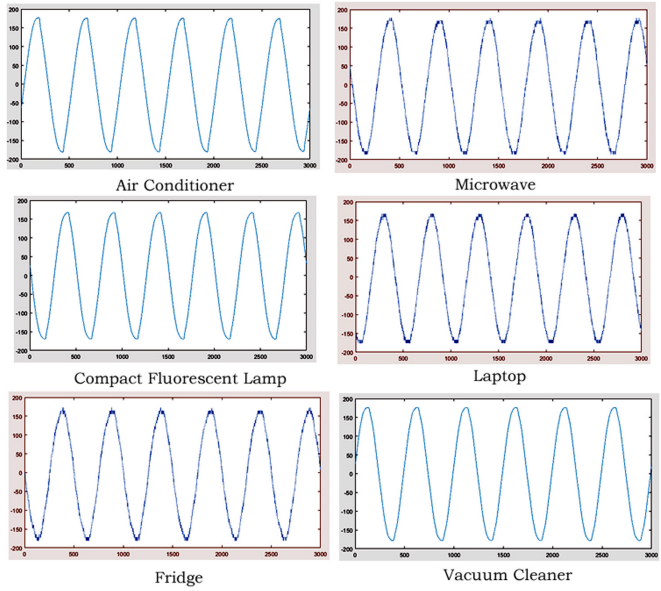

Fig. 2. Voltage signatures of different appliances.

$$x_{rms} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n^{\,2}} \tag{1}$$

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{2}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{n}^{N}\left(x_n - \bar{x}\right)^2} \tag{3}$$

Where x is the input value, n is the sequence number of the data, x_{rms} is RMS, and $\bar{x}$ is mean value.

TABLE III.
EXTRACTED FEATURES

Current		Voltage		Power	
Input	**Value**	**Input**	**Value**	**Input**	**Value**
X_1	Peak to peak	X_{10}	Peak to Peak	X_{19}	Phase angle
X_2	Peak to RMS	X_{11}	Peak to RMS	X_{20}	cosϕ
X_3	I_{RMS}	X_{12}	V_{RMS}	X_{21}	sinϕ
X_4	Minimum	X_{13}	Minimum	X_{22}	Active power
X_5	Maximum	X_{14}	Maximum	X_{23}	Reactive power
X_6	Median	X_{15}	Median	X_{24}	Apparent Power
X_7	Mean	X_{16}	Mean		
X_8	Standard deviation	X_{17}	Standard deviation		
X_9	RMS level	X_{18}	RMS level		

Additionally, we have derived some power parameters by using voltage and current measurements. Phase difference between voltage and current (4), active power (5), reactive power (6), apparent power (7), and active and reactive power factor (8) can be calculated as follows:

$$\varphi = 360.f.\Delta t \tag{4}$$

$$P = V_{RMS}.I_{RMS}.cos\varphi \tag{5}$$

$$Q = V_{RMS}.I_{RMS}.sin\varphi \tag{6}$$

$$S = \sqrt{P^2 + Q^2} = V_{RMS}.I_{RMS} \tag{7}$$

$$cos\varphi = \frac{P}{S} \qquad sin\varphi = \frac{Q}{S} \tag{8}$$

Where φ is the angle between voltage and current, f is the system frequency, Δt is the time delay between signals, P is active power, Q is reactive power, and S apparent power.

B. Deep Neural Network

As stated in Table III, extracted features are very different from each other. They have different ranges and values. BN was used for for both normalization and optimization. Batch Normalization also optimizes the values of all hidden layers along with input values. Layer optimization is crucial for DNN because there are many intermediate layers within the deep network [28]. BN takes mini-batches (9) rather than using all of the data for optimization. This way, it accelerates training and reduces calculation costs by using different normalizaiton terms (10) for each layer (11). Batch Normalization also helps to reduce the internal covariate shift, which is the change of network parameters during training [29].

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{9}$$

$$\rho_B = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - \mu_B\right)^2 \tag{10}$$

$$x_i = \frac{x_i - \mu_B}{\sqrt{\rho_B + \epsilon}} \tag{11}$$

$$y_i = \gamma x_i + \beta \tag{12}$$

As can be seen from (12), BN is not an independent process. It depends on both training results and other mini batches.

$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial y_i}\gamma \tag{13}$$

$$\frac{\partial \ell}{\partial \rho_B} = \sum_{i=1}^{m}\frac{\partial \ell}{\partial x_i}\left(x_i - \mu_B\right)\frac{1}{2}\left(\rho_B + \epsilon\right)^{-\frac{3}{2}} \tag{14}$$

The normalized and shifted values are passed to the other layers according to the chain rule (13) and gradient loss can be calculated through equations (14). The distribution x_i varies between 0 and 1, and each normalized activation is an input to the next layer.

$$\frac{\partial \ell}{\partial \mu_B} = \left(\sum_{i=1}^{m}\frac{\partial \ell}{\partial x_i}.\frac{-1}{\sqrt{\rho_B + \epsilon}}\right) + \frac{\partial \ell}{\partial \rho_B}.\frac{\sum_{i=1}^{m} -2(x_i - \mu_B)}{m} \tag{15}$$

$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial x_i}.\frac{1}{\sqrt{\rho_B + \epsilon}} + \frac{\partial \ell}{\partial \rho_B}.\frac{2(x_i - \mu_B)}{m} + \frac{\partial \ell}{\partial \mu_B}.\frac{1}{m} \tag{16}$$

The gradient of loss l must be back-propagated during the training according to the chain rule (15), (16). This means BN is a differentiable transformation of normalized activation between layers [30].

$$\frac{\partial \ell}{\partial \gamma} = \sum_{i=1}^{m}\frac{\partial \ell}{\partial y_i}.x_i \tag{17}$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{m}\frac{\partial \ell}{\partial y_i} \tag{18}$$

This way model can continue learning on input distribution (17) with less internal covariate shift (18). Where μ_B is the mini-batch mean, ρ_B is batch variance, x_i is the normalized value, and y_i scale and shift, and γ and β are learning parameters.

The back-propagation on DNNs is difficult because of discrete variables and layers. SoftMax transformation is being widely used in neural networks for multi-class classification because of its simplicity and differentiability [31]. The cross-entropy error (19) and SoftMax function (20) update all neurons in the previous layer (21).

$$E\left(t, y\right) = \sum_i t_i.logy_i \tag{19}$$

$$y_i = softmax\left(x_i\right) = \frac{e^{x_i}}{\sum_i e^{x_i}} \tag{20}$$

$$x_i = \sum_i w_{ij}.y_i + b \tag{21}$$

Where t is the target, y is the output value, x is the input for the next neuron, w is the weight, and b is bias.

Multiclassification tasks require SoftMax output, which provides distributional prediction result according to all classes. The partial derivative of the error function for the input layer (22) and hidden layers (23) updates related weights according to the chain rule as can be seen below [32].

$$\frac{\partial E}{\partial w_{ij}} = \sum_i \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial w_{ij}} = \left(y_i - t_i\right)h_j \tag{22}$$

$$\frac{\partial E}{\partial s w_j} = \sum_i \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial h_j}\frac{\partial h_j}{\partial s w_j} = \sum_i \left(y_i - t_i\right)\left(w_{ij}\right)\left(h_j\left(1 - h_j\right)\right) \tag{23}$$

Deep learning is a ML method with a deep multi-layered architecture. Earlier neural networks were shallow networks with fewer layers and limited data utilization. However, recent developments in computational and algorithmic advances have removed the limitations [33-34].

Design of DNNs is relatively difficult due to their complexity and computational load. There are many parameters to be considered such as accuracy, memory usage, calculation count, training time, and power consumption. Deep neural networks are complex nonlinear functions, so it is hard to understand how they work as they combine many methods. Although their internal transparency is low, many different DNNs are being used for various tasks like speech recognition and image classification.

A typical DNN architecture contains many nodes and layers as seen in Fig. 3. Despite the simplicity of classical shallow neural networks, DNNs used to have some limitations such as the normalization problem, calculation cost, and back-propagation. Recent developments in DL have overcome these limitations.

Therefore, a DNN which normalizes data with BN between layers and back-propagated cross-entropy error with SoftMax function has been modelled. The hard tangent function has been used for forward activation because of its low computational cost. Back-propagation and activation functions are really important issues for DNN to achieve the best results.

C. Hardware Implementation

Embedded systems have three layers. The first one is the analog measurement layer, which does all analog measurement via a hall-effect sensor. Then an active filter and current voltage converter board pass analog voltage and current data to the second layer. This layer includes a 32-bit ARM Cortex-M3 microcontroller with 84 MHz clock speed, featuring an onboard 12-bit Analogue-Digital Converter (ADC) and Digital-Analogue Converter (DAC). The third layer is a Compute Unified Device Architecture (CUDA)-enabled embedded computer for artificial intelligence applications. This single board computer has a quad-core ARM Cortex-A57 CPU and Nvidia Maxwell architecture with 128 CUDA cores GPU as computing units, along with 64-bit 4GB onboard LPDDR4 memory.

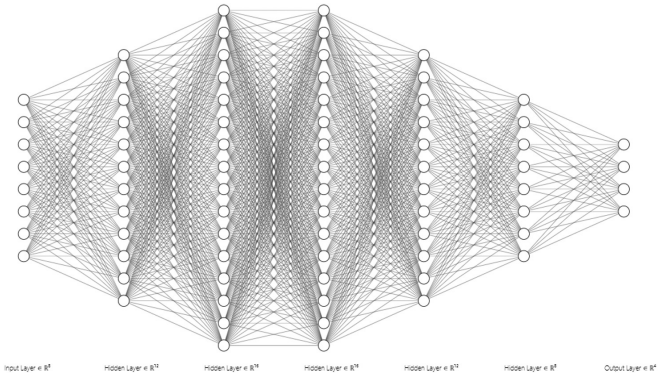

Fig. 3. A typical deep neural network.

Software Pseudocode is as follows:

Loop:

Serial communication → start listening;

Get node identification → NODE;

Start → ADC;

While the ADC timer is on:

ADC →

12-bit 1 kHz;

Get samples → V[n], A[n], n=4096;

End // Features Extraction

Calculate features → X1, X2, X3, X4, X5, X6, X7, X8, X9;

// Peak, RMS, Min, Max, Median, Mean, Deviation for current

Calculate features → X10, X11, X12, X13, X14, X15, X16, X17, X18;

// Peak, RMS, Min, Max, Median, Mean, Deviation for voltage

Calculate features → X19, X20, X21, X22, X23, X24;

// Phase angle, power factor, active, reactive and apparent power

// Deep Neural Network

Import batch parameters → batch_gamma[24], batch_beta[24], momentum;

Calculate → batch_mean[24], batch_var[24];

Reshape and Normalize → Xn[24] = (Xn[24] *batch_mean[24])/(sqrt(batch_var)) *batch_gamma[24] +batch_beta[24];

Import first layer weights and biases → weight1[24:64], bias1[64];

Calculate output → first_layer[64] = Xn[24] * weight1[24:64] + bias1[64];

Activation → out1[64] = max(-1, min(1, first_layer[64]));

Import second layer weights and biases → weight2[64:128], bias2[128];

Calculate output → second_layer[64] = out1[64] * weight2[64:128] + bias2[128];

Activation → out2[128] = max (-1, min (1, second_layer [128]));

Import Third layer weights and bias → weight3[128:256], bias3[256];

Calculate output → third_layer[256] = out2[128] * weight3[128:256] + bias3[256];

Activation → out3[256] = max (-1, min (1, third_layer[256]));

Import Fourth Layer weights and bias → weight4[256:128], bias4[128];

Calculate output → fourth_layer[128] = out3[256] * weight4[256:128] + bias4[128] ;

Activation → out4[128] = max (-1, min (1, fourth_layer [128]));

Import Fifth Layer weights and bias → weight5[128:64], bias5[64];

Calculate output → fifth_layer[64] = out4[128] * weight5[128:64] + bias5[64]; To: Calculate output → fifth_ layer[64] = out4[128] * weight5[128:64] + bias5[64];

Activation → out5[64] = max (-1, min (1, fifth_layer [64]));

Import Sixth Layer weights and bias → weight6[64:16], bias6[16];

Calculate output → sixth_layer[16] = out5[64] * weight6[64:16] + bias6[16];

Activation → out6[16] = (exp(sixth_layer[16])/ sum(exp(sixth_layer[16]));

Classification → PERCENT [index] = max(out6[16]);

All Results → Print (ID [16], Percent [16]);

Device info → Print (NODE, LABEL, ID, PERCENT);

End

Hall effect sensor is a current sensor, but it gives voltage output for use with ADC. Operational-Amplifier (Op-amp) can convert current into voltage and vice versa. This analog data has been sampled and digitized on Arduino for feature extraction. After that, these features have been transferred to Jetson Nano for DNN. The pre-trained DNN has been transferred to Jetson Nano by using the Open Neural Network Exchange (ONNX) model and Python libraries. Results can be logged on the device or sent to another system via the Internet. An embedded display can also be used for real-time monitoring.

The block diagram of the DNN used for appliance identification using voltage and current signal signatures can be seen in Fig. 4. ONNX model is also provided, which is useful for hardware implementation. The pseudocode of the software is also given below. Hardware layers of the embedded system can be seen in Fig. 5.

III. RESULTS

The implanted DNN has been tested with the PLAID dataset and a comparison of different architectures can be seen in Table IV. We have used Neural Network Console [35] on a PC with a 4.2 GHz 8-core processor and 16 GB 3200 MHz RAM with GPU with 1920 CUDA cores. This console is powered by Python neural network libraries and supports CUDA. All architectures have 24 inputs and

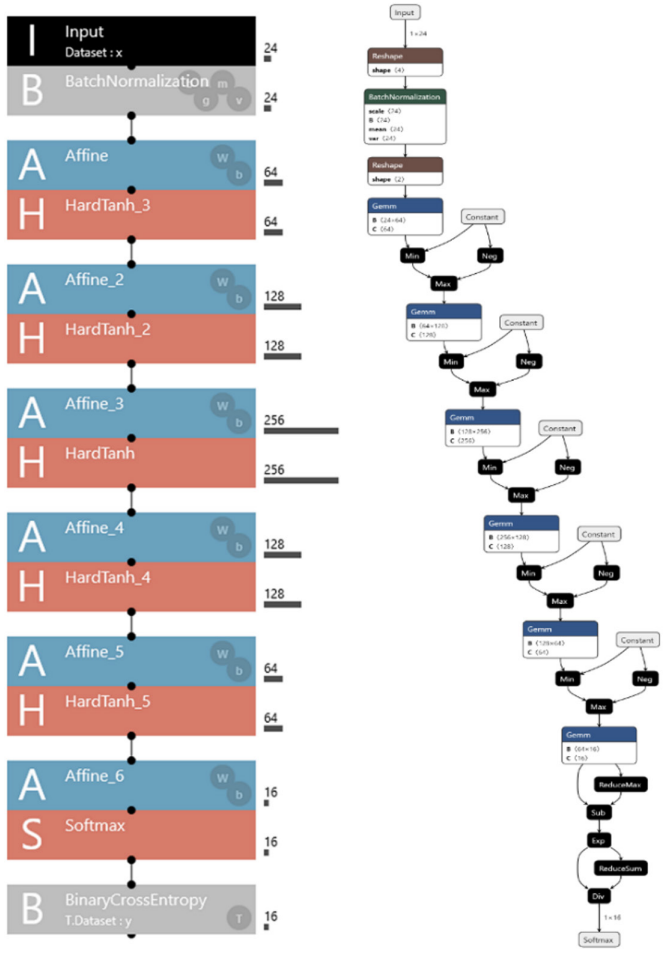

Fig. 4. Block diagram of the deep neural network and open neural network exchange model [35].

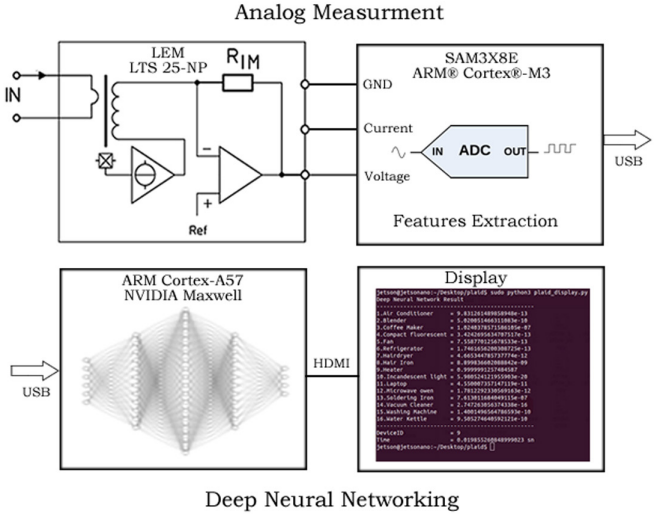

Fig. 5. Block diagram of the embedded system.

TABLE IV.
COMPARISON OF DIFFERENT DEEP NEURAL NETWORK ARCHITECTURES

DNN Architecture	Training Error	Validation Error	Total Accuracy	Total Precision	F1 Score
64-128-256-128-64	0.00900	0.00808	0.980	0.9869	0.966
32-64-128-64-32	0.01012	0.00916	0.970	0.9800	0.958
64-128-256-64-32	0.01018	0.00883	0.976	0.9831	0.962
32-64-128-64-32	0.01059	0.01098	0.971	0.9799	0.957
32-64-128-256-64	0.00873	0.00816	0.972	0.9817	0.960

16 outputs, but the number of hidden layers is different. The 5 best resulting DNN models can be seen in Table IV. Training and validation errors, training time, accuracy, precision, recall, and F measure values have been compared.

The best F score was 0.9666 with 0.9869 accuracy in the test. We have tried more complex DNN models; however, no improvement has been observed. The confusion matrix can also be also seen in Fig. 6 F score and recall values are good enough except for the blender (y1) because there are only two samples in the dataset related to this class. Other than that, the results are promising.

Additionally, this study has shown how BN is crucial for the DNN learning process. That means all layers must be normalized for DNN, rather than normalizing the input only. Another problem is back-propagations when it comes to DNNs. It is common to use the SoftMax output layer for multiclassification problems. The output error must be distributed over hidden layers elegantly. SoftMax back-propagation with cross-entropy is a useful technique to do this. The learning curve of the DNN is shown in Fig. 7. This curve represents the change of cost, training error, and validation error based on each epoch.

Moreover, the effects of different activation functions were dramatical. Conventional functions such as tangent hyperbolic and sigmoid cannot handle DNNs. The sigmoid activation function gives a smoother learning curve, but training and validation error values increase. This is because the sigmoid function suppresses negative values. On the other hand, the tangent hyperbolic function can be used instead, but this time the calculation cost will rise along with training error ripple. Therefore, the hard tangent hyperbolic function has been preferred. A hard saturating version of an activation

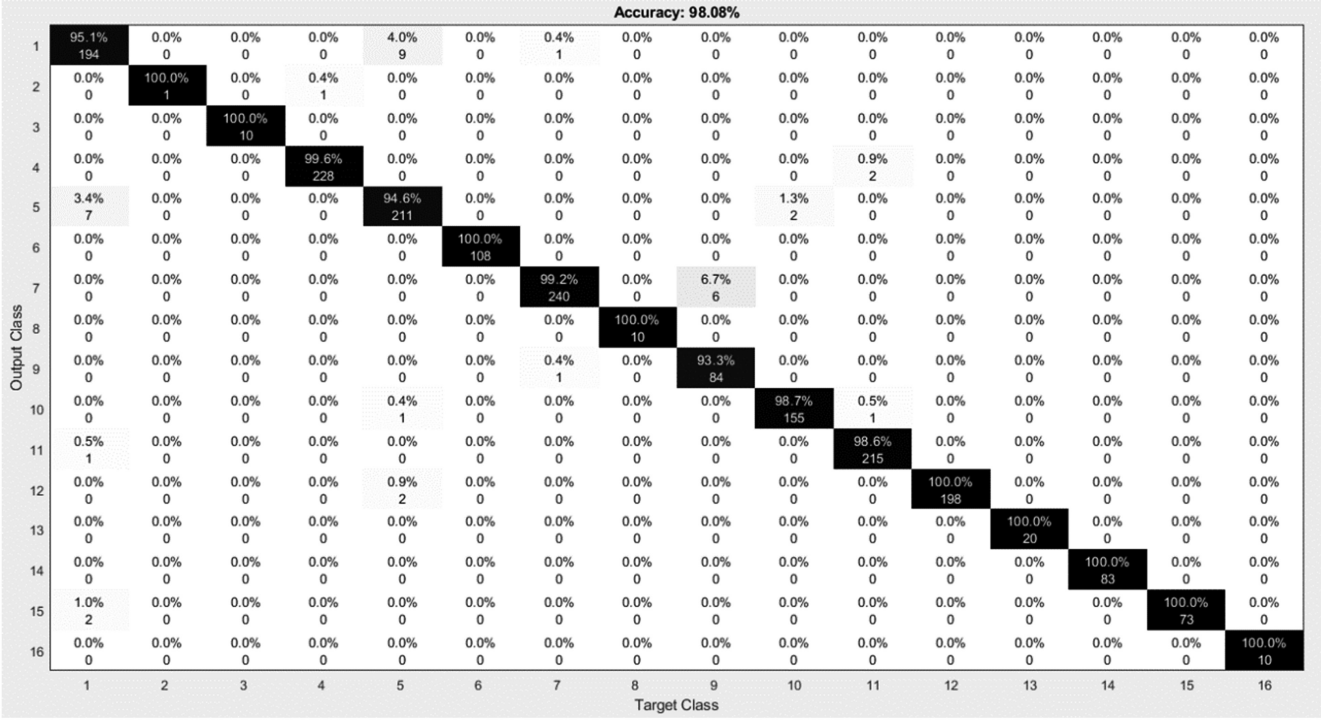

Fig. 6. Confusion matrix.

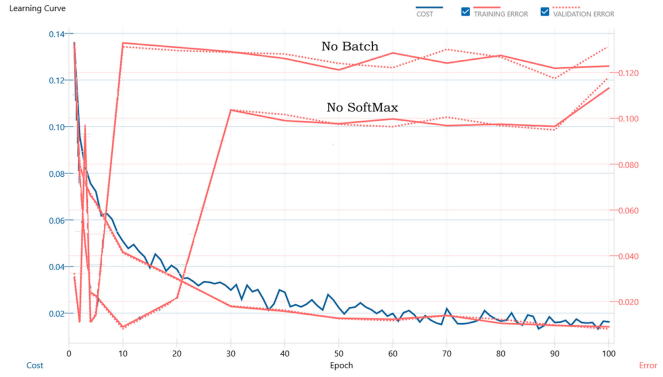

Fig. 7. Learning curve.

function can be constructed by taking a first-order Taylor expansion from zero to an appropriate range [36].

Our system is using a hall-effect sensor which is non-intrusive for analogue measurement. The device measures both voltage and current from a single sensor and extracts features at the same time. Then a pre-trained DNN analyzes these features to identify the appliance. In this way, the appliance is monitored and identified by the system. The accuracy of the system is dependent on big data, so this database can be improved by introducing new devices.

This method includes the utilization of a neural network on an embedded system with high performance and accuracy. The real contribution of the paper is the realization of the method and the development of a single sensor intelligent measurement device. Moreover, the dataset used to train DNNs has multiple aspects, so right approach is needed for better data utilization. Showing how different DNN architectures can affect the result is also another contribution of this paper.

Results screen captures of some appliances are given in the figures 8 to 13. This data can be retrieved through the internet via TCP/IP, as they can be observed on the device terminal window. Furthermore, these data can be logged into a database to improve the accuracy of the system. The device can collect all current and voltage measurement data from intelligent power outlets and identify the type of appliance. The system gives percentages of probabilities according to the device ID. It can be seen from the device screen capture that some appliances have higher probility than others. The reason for that is the database, which has different number of samples from various appliances. That is why the system has been designed for user who can add different appliances into the database later. This way, the system will be improved continuously.

V. DISCUSSION

Deep learning technologies have been rapidly evolving in the past decade. This is also bringing some problems on both the software and hardware sides. Training and testing of DNNs require powerful hardware with GPU support and high-speed computers with fast data storage. On the other hand, complex calculations and statistical

```
jetson@jetsonano:~/Desktop/plaid$ sudo python3 plaid_display.py
Deep Neural Network Result
-------------------------------------------------
1.Air Conditioner      = % 0.0017616770691210291
2.Blender              = % 1.2923767199624472e-07
3.Coffee Maker         = % 0.014183181844259067
4.Compact fluorescent  = % 4.40340621465664e-07
5.Fan                  = % 98.76212537752299
6.Refrigerator         = % 0.0001299153388546633
7.Hairdryer            = % 3.844486667906239e-09
8.Hair Iron            = % 2.147856621438854e-09
9.Heater               = % 3.726772759839552e-08
10.Incandescent light  = % 3.751262212986629e-08
11.Laptop              = % 7.90428739107951e-05
12.Microwave owen      = % 0.0414034472389369
13.Soldering Iron      = % 1.180316645601061
14.Vacuum Cleaner      = % 9.72982879472609e-09
15.Washing Machine     = % 2.4108217439035956e-09
16.Water Kettle        = % 5.001922687265507e-08
-------------------------------------------------
DeviceID               = 5
Accuracy               = % 98.76212537752299
Time                   = 0.0018076896667480469 sn
jetson@jetsonano:~/Desktop/plaid$
```

Fig. 8. Results screen captures of the fan.

methods are being developed to overcome the problem of handling this big data. It is all about processing the data in the most efficient way to get what you need from it.

This study has focused on appliance identification problems by using only simple line parameters such as voltage and current. There are many imbalanced datasets on this problem but there are few innovative approaches. Multi-state household appliance identification [37] uses non-intrusive load monitoring with an ANNwith 89% accuracy. Automated plug-load identification [38] achieves 86% accuracy by using ML with an ensemble classifier. Handling imbalance in an extended PLAID [39] achieves better result with 94.5% accuracy by using a support vector machine. As can be seen, those conventional methods such as ML and ANNs do not achieve good results. However, the DL methods used in this study achieve relatively better results with 98% accuracy in this study.

```
jetson@jetsonano:~/Desktop/plaid$ sudo python3 plaid_display.py
Deep Neural Network Result
-------------------------------------------------
1.Air Conditioner      = % 2.5833283181631 36e-141
2.Blender              = % 1.3341359341489982e-62
3.Coffee Maker         = % 3.589863387591776e-180
4.Compact fluorescent  = % 3.1727763018350224e-59
5.Fan                  = % 1.926441740045761e-130
6.Refrigerator         = % 2.90524927077692e-157
7.Hairdryer            = % 8.157921552574418e-104
8.Hair Iron            = % 3.720533498727688e-127
9.Heater               = % 7.966736104112201e-103
10.Incandescent light  = % 100.0
11.Laptop              = % 1.1655873188781286e-50
12.Microwave owen      = % 7.873775913212049e-113
13.Soldering Iron      = % 6.748343385837354e-109
14.Vacuum Cleaner      = % 3.034504206633392e-151
15.Washing Machine     = % 1.1640443484999129e-97
16.Water Kettle        = % 5.160210745735886e-154
-------------------------------------------------
DeviceID               = 10
Accuracy               = % 100.0
Time                   = 0.004964590072631836 sn
jetson@jetsonano:~/Desktop/plaid$
```

Fig. 9. Results screen captures of the incandescent light.

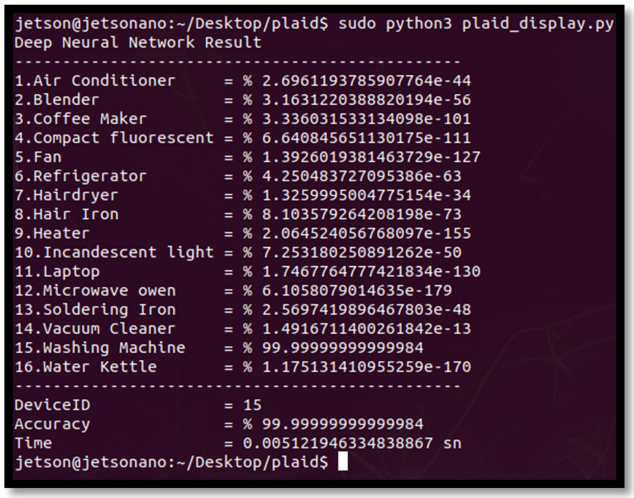

Fig. 10. Results screen captures of the washing machine.

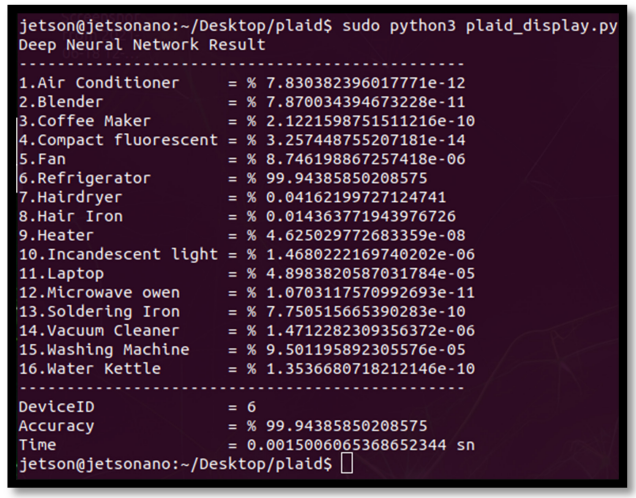

Fig. 12. Results screen captures of the refrigerator.

There are many different DL methods and models. These methods and models require complex calculations, but they also need to be less exhausting on hardware. For this reason, different normalization methods with effective activation functions should be considered for DNNs. Back-propagation is another problem for multi-layered DNNs. Combining more than one technique is very useful for creating effective and elegant new methods. This study has created a DNN by using BM and SoftMax back-propagating with cross-entropy for the identification of appliances from an imbalanced dataset.

VI. CONCLUSION

This developed method has been tested in real-time by running on an embedded artificial intelligence computer. First, the test data was checked to determine whether the algorithm worked correctly or not, and then the device detection was carried out with real-time measurement values. By increasing the number of modules used by the device for analog measurement, the information on energy-consuming devices in the whole system can be collected and analyzed via smart power outlets. It is also possible for users to introduce different devices that are not in the dataset to the system by labeling them. In this way, a constantly learning system has been developed within a constantly updated database. In addition, the device recognition diversity of the system can be improved by making use of different data sets such as those given in the introduction.

Future studies will focus on the implementation of this system for a smart grid or smart home. This way, it will be easier to manage power consumption and remote control of all appliances. Especially, it is good for use as an Internet of things (IOT)-based system. On the other hand, this kind of intelligent measurement device will help renewable energy systems to be optimized for power consumption.

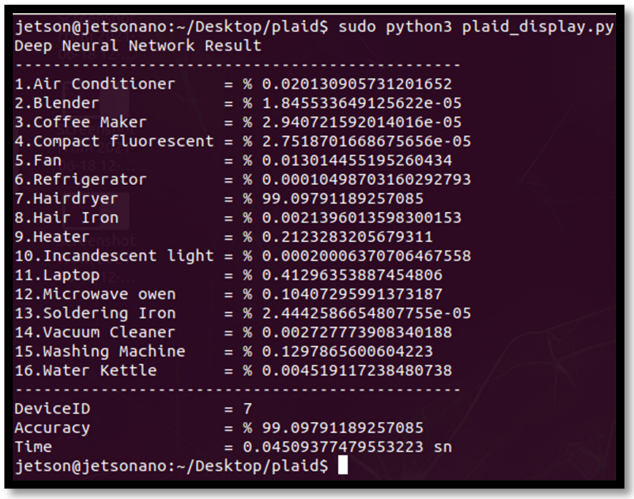

Fig. 11. Results screen captures of the hairdryer.

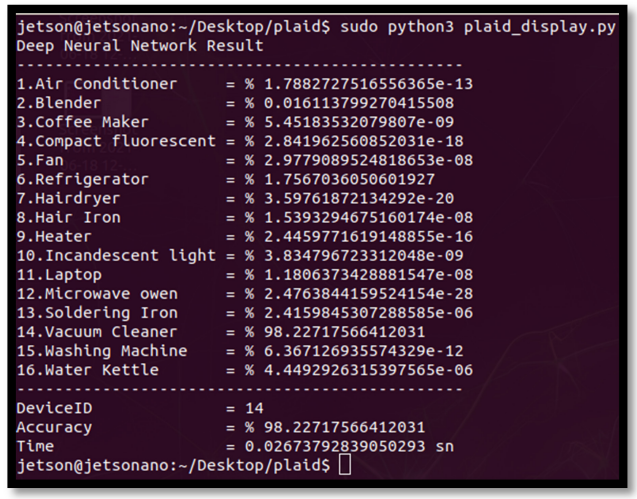

Fig. 13. Results screen captures of the vacuum cleaner.

Availability of Data and Materials: The data that support the findings of this study are available on request from the corresponding author.

Peer-review: Externally peer-reviewed.

Author Contributions: Concept – Y.G.; Design – Y.G.; Supervision – Y.G.; Resources – Y.G.; Materials – Y.G.; Data Collection and/or Processing –Y.G.; Analysis and/or Interpretation – Y.G.; Literature Search – Y.G.; Writing – Y.G.; Critical Review – Y.G.

Acknowledgments: This study has been conducted with personal funds, and facility of Kırklareli University have been used.

Declaration of Interests: The author has no conflicts of interest to declare.

Funding: This study received no funding.

REFERENCES

1. R. Medico et al, "A voltage and current measurement dataset for plug load appliance identification in households," *Sci. Data*, vol. 7, no. 1, p. 49, 2020. [CrossRef]
2. T. Picon, M. N. Meziane, P. Ravier, G. Lamarque, C. Novello, J. C. Le Bunetel, Y. Raingeaud, "COOLL: Controlled On/Off Loads Library, a Public Dataset of High-Sampled Electrical Signals for Appliance Identification", https://arxiv.org/abs/1611.05803 2016.
3. A. Ridi, C. Gisler, and J. Hennebert, "ACS-F2 – A new database of appliance consumption signatures," in *Soft Comput. Pattern Recognit. (SoCPaR)* 6th International Conference of, vol. 2014, 2014, pp. 145–150. [CrossRef]
4. J. Kelly, and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Sci. Data*, vol. 2, p. 150007, 2015. [CrossRef]
5. D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study," *Sci. Data*, vol. 4, p. 160122, 2017. [CrossRef]
6. J. Zico Kolter, and M. J. Johnson, "REDD: A public dataset for energy disaggregation research," 2011, San Diego, CA, USA, SustKDD. Available: http://redd.csail.mit.edu/.
7. O. Hamid, M. Barbarosou, P. Papageorgas, K. Prekas, and C.-T. Salame, "Automatic recognition of electric loads analyzing the characteristic parameters of the consumed electric power through a Non-Intrusive Monitoring methodology." *Energy Procedia*, vol. 119, pp. 742–751, 2017. [CrossRef]
8. 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," 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Boston, MA, 2010, 2010, pp. 1–9. [CrossRef]
9. A. Ridi, C. Gisler, and J. Hennebert, "Appliance and state recognition using Hidden Markov Models," International Conference on Data Science and Advanced Analytics (DSAA), Shanghai, 2014, 2014, pp. 270–276. [CrossRef]
10. I. MPAWENIMANA, A. PEGATOQUET, W. T. SOE, and C. BELLEUDY, "Appliances identification for different electrical signatures using moving average as data preparation," Ninth International Green and Sustainable Computing Conference (IGSC), Pittsburgh, PA, USA, 2018, 2018, pp. 1–6. [CrossRef]
11. O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, A. M. Umar, O. U. Linus, "Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition," in *IEEE Access*, vol. 7, pp. 158820–158846, 2019. [CrossRef]
12. A. B. Slama, Ł. Lentka, A. Mouelhi, M. Diouani, M. F. Sayadi, and J. Smulko, "Application of Statistical Features and Multilayer Neural Network to Automatic Diagnosis of Arrhythmia by ECG Signals," *Metrology and Measurement Systems*, vol. 25, no.1, pp. 87–101, 2018.
13. S. Sadeghi, W. A. MacKay, R. M. van Dam, and M. Thompson, "Algorithm for real-time detection of signal patterns using phase synchrony: An application to an electrode array," *Meas. Sci. Technol.*, vol. 22, no. 2, pp. 1–12, 2011. [CrossRef]
14. S. Khokhar, A. A. Mohd Zin, A. P. Memon, and A. S. Mokhtar, "A new optimal feature selection algorithm for classification of power quality disturbances using discrete wavelet transform and probabilistic neural network," *Measurement*, vol. 95, pp. 246–259, 2017. [CrossRef]
15. K. Lee, L. Huchel, D. H. Green, and S. B. Leeb, "Automatic power frequency rejection instrumentation for nonintrusive frequency signature tracking," in *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021. [CrossRef]
16. J. Cui, G. Shi, and C. Gong, "A fast classification method of faults in power electronic circuits based on support vector machines," *Metrol. Meas. Syst.*, vol. 24, no. 4, pp. 701–720, 2017. [CrossRef]
17. Z. Moravej, M. Pazoki, and M. Khederzadeh, "New pattern-recognition method for fault analysis in transmission line with UPFC," in *IEEE Trans. Power Deliv.*, vol. 30, no. 3, pp. 1231–1242, 2015. [CrossRef]
18. P. Pawar, M. TarunKumar, and K. Panduranga Vittal, "An IoT based Intelligent Smart Energy Management System with accurate forecasting and load strategy for renewable generation," *Measurement*, vol. 152, pp. 107–187, 2020. [CrossRef]
19. T. Kurczveil, P. Diekhake, J. Liu, and E. Schnieder, "Consumer load measurement in automated buildings," *Measurement*, vol. 51, pp. 441–450, 2014. [CrossRef]
20. W. L. Rodrigues, F. A. S. Borges, A. F. S. Veloso, R. A. L. Rabêlo, and J. J. P. C. Rodrigues, "Low voltage smart meter for monitoring of power quality disturbances applied in smart grid," *Measurement*, vol. 147, pp. 1–15, 2019. [CrossRef]
21. M. G. Xibilia, M. Latino, Z. Marinković, A. Atanasković, and N. Donato, "Soft sensors based on deep neural networks for applications in security and safety," in *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 7869–7876, 2020. [CrossRef]
22. K. Organisciak, and J. Borkowski, "Single-ended quality measurement of a music content via convolutional recurrent neural networks," *Metrol. Meas. Syst.*, vol. 27, no. 4, pp. 721–733, 2020. [CrossRef]
23. F. Ciancetta, G. Bucci, E. Fiorucci, S. Mari, and A. Fioravanti, "A new convolutional neural network-based system for NILM applications," in *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021. [CrossRef]
24. G. Shi, Y. He, and C. Zhang, "Feature Extraction and Classification of CATA luminescence Images Based on Sparse Coding Convolutional Neural Networks," in *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021. [CrossRef]
25. H. B. Zhuo, F. Z. Bai, and Y. X. Xu, "Machine vision detection of pointer features in images of analog meter displays," *Metrol. Meas. Syst.*, vol.27, no. 4, pp. 589–599, 2020. [CrossRef]
26. J. Sun, C. Yan, and J. Wen, "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning," in *IEEE Trans. Instrum. Meas.*, vol. 67, no. 1, pp. 185–195, 2018. [CrossRef]
27. A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "A novel feature extraction for robust EMG pattern recognition," *J. Comput.*, vol. 1, no. 1, pp. 71–80, 2009. Available: https://arxiv.org/abs/0912.3973.
28. J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization", 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada. Available: https://arxiv.org/abs/1806.02375.
29. J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016. United States of America: Cornell University. Available: https://arxiv.org/abs/1607.06450.

30. S. Ioffe, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015. United States of America: Cornell University. Available: https://arxiv.org/abs/1502.03167.
31. A. F. T. Martins, and R. Fernandez Astudillo, "From softmax to Sparsemax: A sparse model of attention and multi-label classification," 2016. United States of America: Cornell University. Available: https://arxiv.org/abs/1602.02068.
32. P. Sadowski, "Notes on backpropagation." United States of America: University of California Irvine. Available: https://www.ics.uci.edu/~pjsadows/notes.pdf. .[Accessed: 7.06.2020].
33. A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," 2017. United States of America: Cornell University. Available: https://arxiv.org/abs/1605.07678.
34. I. D. Dinov, "Deep learning, neural networks," In *Data Science and Predictive Analytics*. Cham: Springer, 2018. [CrossRef]
35. *Neural Network Console by Sony*. Available: https://dl.sony.com/. .[Accessed: 23.04.2020].
36. C. Gulcehre, M. Moczulski, M. Denil, and Y. Bengio, "Noisy activation functions," 2016. United States of America: Cornell University. Available: https://arxiv.org/abs/1603.00391.
37. Y. Zhang, B. Yin, Y. Cong, Z. Du, and M. S. Household, "Multi-State Household Appliance identification based on convolutional neural networks and clustering," *Energies*, vol. 13, no. 4, p. 792, 2020. [CrossRef]
38. J. Gao, E. C. Kara, S. Giri, and M. Bergés, "A feasibility study of automated plug-load identification from high-frequency measurements," IEEE Global Conference on Signal and Information Processing (GlobalSIP), Orlando, FL, 2015, pp. 220–224. [CrossRef]
39. L. De Baets, C. Develder, T. Dhaene, D. Deschrijver, J. Gao, and M. Berges, "Handling imbalance in an extended PLAID," *Sustain. Internet ICT Sustain. (SustainIT), Funchal*, vol. 2017, pp. 1–5, 2017. [CrossRef]