Large Scope Device Recognition by Power Usage for Crownstones

An investigation of existing methods and their limits

Merijn van Tooren

ICA-3689557
Department of Information and Computer Sciences
Utrecht University
Netherlands
August 19, 2019

Abstract

Electrical appliance classification has great potential. It has potential uses in analysis of power usage within households, automation of households, and detection of hazards and electrical decay. The field is well researched, but has yet to see successful mass deployment in the modern world. A new device called the Crownstone may be the solution, as it is capable of intrusive load monitoring and being developed to be distributed into many households. In this research, an easily implementable established method for classification of electrical appliances by intrusive load monitoring is tested on a new, more challenging dataset recorded using Crownstones. An analysis is made of the achievable accuracy, as well as the effects of the noise and larger number of classes. It is found that the method continues to perform surprisingly well under these more demanding conditions, especially with the help of simple preprocessing steps.

Contents

List of Figures

List of Tables

1 Introduction

Technology is everywhere. A typical modern household involves dozens of electrical appliances, each fulfilling one or more distinct functions. In recent years, "Smart Home" technology and the "Internet of Things" have been on the rise, becoming more and more prevalent. These are technologies that may provide security, convenience, or new possibilities to users, e.g. by automating certain procedures like turning lights on and off, offering voice interfacing, or by recognising certain dangers. Solid contemporary examples are the Amazon Alexa and Homey.

One subset of such technology is that which is concerned with monitoring power usage. Already many homes have been or are being equipped with "Smart Meters", which improve the user's awareness of power consumption in their household. In order to provide an optimal readout, it would be functional to separate these statistics by appliance and label them accordingly, allowing the user to deduce from one glance which devices are playing important roles in their electricity bills.

1.1 Device Recognition by Power Usage

This research concerns the recognition of electrical appliances by applying algorithms to measurements obtained through low-frequency Intrusive Load Monitoring. The goal is to be able to classify appliances by type, e.g. "Refrigerator" or "Bean-to-Cup Coffee Machine".

Intrusive Load Monitoring, henceforth ILM, entails the measurement of appliances' power usage individually, within a building's grid, which is what makes it "intrusive". This is as opposed to Non-Intrusive Load Monitoring, henceforth NILM, which is less intrusive by virtue of the measurements being taken at the main supply, observing only an aggregate of all loads within the building.

This research specifically deals with low-frequency ILM, which is to say that there is no more than one measurement per appliance per second. In other words, available data has a frequency lower than 1 Hz. This notably excludes the possibility of waveform analysis, and is more suited to analysis of daily usage patterns as well as the patterns and levels of power usage appliances can exhibit.

Truly, many experiments have already been done within this field, as well as with high-frequency recordings and NILM. Generally, they report high accuracy ratings between 85 and 100 percent. These are serviceable accuracies for consumer purposes. It would be desirable to see such recognition implemented in smart homes today, but there are factors holding this back. Two are of primary concern in this research: availability and capability.

Especially with ILM, availability is an obstacle. It is likely to be costly to place sensors on all home appliances, and disruptive to residents. The Crownstone [1] offers a solution to this, and will be further explored in subsection 1.3.

In terms of capability, there is an issue with the number of classes. Most of related researches have only attempted to distinguish between 5 to 30 different classes of appliance. This does not translate to a consumer application. Antonio Ridi et al. [12] survey existing papers within the field, and conclude that there is more to be learned from larger, more complete datasets.

One or more methods of appliance recognition must be verified for larger numbers of classes, to advance towards a feasible consumer application of such recognition methods. Therefore, this research will take a simple method with a high accuracy rating from another paper [9], and apply it to a much broader dataset.

The method in the referenced paper uses features extracted from 24-hour samples of appliances, with mean power usage recorded every few (3-5) seconds. These features are used in simple machine learning algorithms offered by the publicly available Weka toolkit.

In many of the related works, the Weka toolkit [3] is used to run classification algorithms. Because of this prevalence, this research will also use this toolkit.

1.2 Power Usage Data Collection

There are already several publicly available datasets from different low- and high-frequency ILM and NILM experiments. However, at the time of writing, none fulfil the requirements of this research.

A new, large dataset has to be recorded, featuring full 24-hour samples of each device, with mean power usage recorded every few (3-5) seconds, and a significantly larger number of classes than in other studies.

1.3 The Crownstone Technology

The classification task is provided by the Crownstone project, by the company DoBots. The Crownstone is a small item of electronics, designed to be fitted behind wall sockets, or as an intermediary plug between appliance and socket. In this way, a Crownstone is introduced between a single device and the building's grid. A Crownstone can measure the voltage over, and the current going through the device, in real time (Sample Rate: 6kHz). Noise accounts for about 3 W of uncertainty in resulting power usage statistics. This data can be used to classify the device. The Crownstone has more features and applications, which can be found in more detail at [1]. The goal of this research is to explore the feasibility of electrical device recognition using Crownstones.

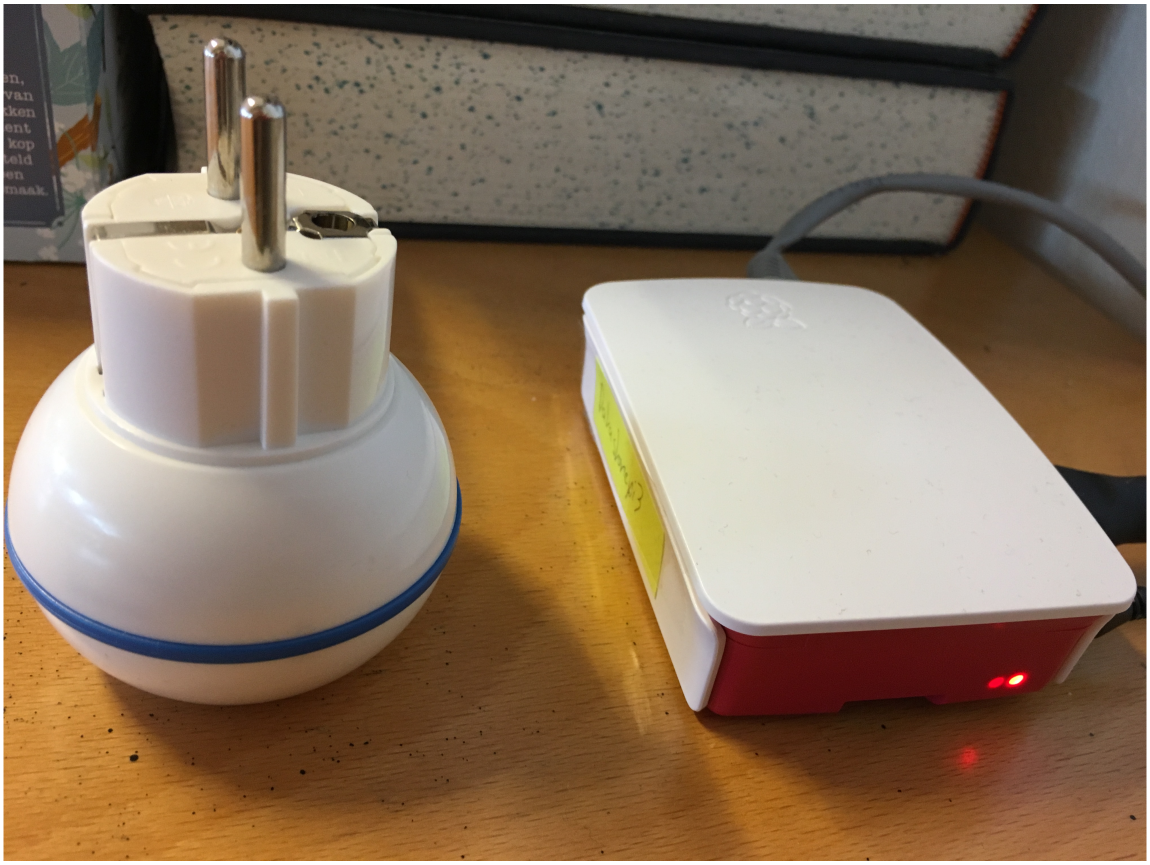


Figure 1: Crownstone Plug and Raspberry Pi.

2 Related Works

In this section, related works in the field of device recognition are studied to give background to this research. Furthermore, an in-depth look at research done by Reinhardt et al. [10] [9] serves as main reference for the experiments performed in this research.

2.1 General Study

A survey by Ridi et al. [12] was made on ILM and device recognition, in which all existing methods were compared. In this survey, Ridi et al. conclude that no "leading" or preferred method for device recognition can be identified. Furthermore, Ridi et al. find that the selection of device type classes used in researches shows great variability, and many device types do not show up regularly.

A considerable number of papers reporting experiments with device recognition

Paper	Rate	Res	Trace Size	Classes	Algorithm	Accuracy
[10]	1.6kHz	16b	320ms	15	BN	100%
[9]	1Hz	NM	24h	31	RC	95.5%
[2]	96kHz	20b	10 periods	9	RF	99.8%
[6]	NM	NM	20ms	14	HC	NA
[13]	0.1Hz	NM	1h	5	k-NN, GMM	85%
[11]	0.1Hz	NM	1h	10	GMM	93.6%
[5]	NM	NM	1 period	25	SVM	99.8%
[8]	1/120Hz	NM	100 samples	8	NN	95.26%

NM: Not Mentioned NA: Not Applicable

Table 1: Table comparing related works by sampling rate, sample resolution, size of intervals recorded, number of classes, classification algorithm used and accuracy reported.

and ILM was found. See Table 1 for an overview. The table only lists data for the best results of each paper. Rate stands for the sampling rate, Res stands for sample resolution, or the number of bits per sample, Trace Size stands for the amount of samples used in one classification, Classes stands for the number of device types distinguished between, Algorithm stands for the algorithm used in the best results and Accuracy stands for the accuracy score of those results. Accuracy is omitted in [6] because the classification is not an identification of the device types listed, and is rather a result of clustering. It cannot be compared in terms of accuracy. There is a large variation in several of these attributes, e.g. sampling rate, is, and how many authors have omitted the sample resolution, which may be an instrumental factor in classification accuracy, especially when low levels of power consumption are involved.

In [6], in an early research in 2007, Lam et al. try something relatively unconventional. Rather than predetermining a list of device types for classification, they apply hierarchical clustering. Their features are features describing the shape of the V-I trajectory of a trace of 20ms. With their clustering, they find a different subdivision into classes, which make more sense from the perspective of the classifier.

Kato et al. [5] also perform early studies within this field, managing an accuracy of 99.8% on recognition of 25 device types as well as a 95.8% accuracy on recognising a small number of individual device types, using a Support Vector Machine. Their features are inner products of base vectors with segments of the measured current waveforms.

Two papers [13] [11] detail the investigation of k-Nearest Neighbours and Gaussian Mixture Models as methods for device recognition. The former [13], listing only 5 device types, reports a 85% accuracy with both algorithms. The latter research [11] utilises the same sampling rate and trace size, as well as twice as many device classes, but different, less common features akin to derivatives, and scores much higher, achieving an accuracy of 93.6% with Gaussian Mixture Models.

Paradiso et al. [8] experiment with an Artificial Neural Network for the classification of 8 device types, with an unconventional choice of features. They use thresholds to count samples within power ranges and transitions of certain magnitudes, and these counts supplement a few more common statistical features. An accuracy of 95.26% is reached in this research.

Englert et al. make use of the Weka toolkit in [2] in their investigation of accuracy based on algorithm and feature extraction. They experiment with the highest sampling rate, namely 96kHz. A high accuracy is achieved on a low number of device types, but the greater achievement of this work is that they also show high scores in classifying specific device models, as well as device modes. Specifically, they report 100% accuracy in classifying between monitors, on which they focused for this part of the experiment, and they present a very respectable confusion table concerning mode classification of chargers, switches and monitors, with 3, 5 and 6 modes respectively.

Englert et al. do not only attempt to classify device types and models, but also experiment with mode classification. They invest heavily in their uniquely high sampling rate, using the individual Root Mean Square (RMS) values of exactly 800 harmonics as features, in addition to the added RMS values of even, as well as the added RMS values of odd harmonics from this group. Real power, phase shift, and crest factor (the ratio of the current waveform's peak to its RMS value) are also features. Without the special even and odd RMS sums, accuracy on device types is 99.6%, and those two extra features raise this result to 99.8%, cutting the remainder in half.

However, when attempting to classify between device models of monitors, accuracy drops to 87%. The authors remedy this by creating a separate Random Forest classifier for this subtask, using Greedy Stepwise feature ranking to select 20 of their features for use in this classifier. It proves highly successful, raising the accuracy to 100% for this subtask. This success reinforces the concept of creating a tree of specialised classifiers for classification subtasks. The researchers go on to attempt classification of operational modes of devices, investigating the effectiveness of their method on three separate cases: smartphone charger, network switch and TFT monitor. On the charger, a full 100% accuracy is achieved in recognising the modes idle, loading and done. In the case of the switch, the method is capable of flawlessly detecting the number of wired connections. Finally the method can recognise the difference between a fully white screen and a fully black screen in the case of the monitor, but when the differences become more detailed or hue-related, accuracy drops. This shows that specialised classifiers will likely be able to easily and accurately recognise operational modes of devices.

2.2 Main Reference

Reinhardt et al. [10] compare the effectiveness of a number of algorithms offered by the Weka toolkit with regard to device type classification, using high frequency data at 1.6kHz, distinguishing between a total number of 16 device type classes, comparing 9 different classification algorithms, namely Bagging, Bayesian Network, J48, JRip, LogitBoost, Naive Bayes, Random Committee, Random Forest and Random Tree. No particular explanation is given for this specific selection.

By means of an FFT, harmonic features are extracted from the high frequency current measurements taken from the 'inrush' and 'steady state', that is to say, immediately after activation, and 5 seconds later, respectively. The magnitude of the fundamental frequency, the first four odd harmonics, and the DC component are the features taken from the FFT, and these are used in addition to the RMS and arithmetic mean of the current, as well as the maximum current and the phase shift. By this method, very high accuracies are achieved, the best of which is a 100% accuracy achieved by 25-fold cross validation using a Bayesian Network classifier.

Following this, Reinhardt et al. investigate classification using the Weka toolkit on low frequency data in [9]. Plugwise devices are used to gather samples at a theoretical rate of 1Hz, although there are frequent gaps of one or more seconds in the public version of the dataset. In this case, power usage is recorded, rather than current, and in theory this makes little difference as power is the product of current and tension, and the tension on the grid is very stable. 33 device types are monitored for samples totalling between 10 and 100 days of data per device. Each day of measurements, from midnight to midnight, is a sample for classification, and is named a 'trace'. In order to compensate for the low frequency and larger number of classes, and to explore many possibilities, 512 features are implemented and extracted for use in classification. Following is a summary of this large quantity of features.

By manually choosing an activity threshold, "activity intervals" are defined as periods in which the measured power usage is higher than the activity threshold. Closely grouped activity intervals are then named "usage blocks". Several features are gained by analysing the frequency and duration of these periods. The amount of time in which the power usage measured is within 2% of the maximum, the minimum, and the average power usage are also features. A similar total duration is calculated for the time in which the power usage measured is within any of 10 sections of 10% of the total range between minimum and maximum power usage.

Furthermore, the day is divided into complete sets of periods of 10 minutes, 1 hour, 2 hours, 6 hours and 12 hours. These periods, in addition to the full day, the longest activity interval and the total activity time, are statistically processed for features such as total, mean, median and maximum power usage each.

The amount of times the power usage changes by more than 5% and more than 60% are counted, as well as the amount of times certain set thresholds in Watts are exceeded, and threshold excesses shorter than 20 seconds are counted differently from longer ones.

Peaks are analysed for the steepness of their slopes, their frequency of occurrence, and their magnitudes. Noise is measured by applying a low pass filter to the trace and comparing its result with the original. A discrete FFT is also used, generating a feature for each of its ten frequency bands.

This is not a complete summary of all 512 features, but it covers the gross majority, and most other features are related to these analyses.

Using these features, Reinhardt et al. report an accuracy of 95.5% using a Random Committee classifier. The Bayesian Network classifier did not perform nearly as well, with an accuracy of 91.48%. It also takes much longer to train a Bayesian Network as opposed to a Random Committee.

The report provides an information gain analysis of the features, along with the most important 15 features ranked by relevance. Most important are the features reporting the maximum and average power usage during the day and specifically during activity, and it seems that this mean and max power usage are the most distinguishing aspects of electrical devices when measured over a long time at a low sampling rate.

2.3 Summary

In summary, regarding Table 1, it is possible to formulate some rudimentary theories. Firstly, it would seem that experiments with waveform-level features have the highest accuracies (100%, 99.8%, 99.8%), even when compared to low-rate experiments which use large amounts of data (up to 24 hours of samples). Secondly, it is difficult to identify characteristics of algorithms that produce better scores, and experiments using the same algorithms have greatly varying results (e.g. [13] and [11]). This potentially implies that the choice of algorithm is of much lesser importance, however comparisons within researches (e.g. [10]) contest this theory, and it is unknown how algorithms will compare under more taxing requirements. It would also seem that the number of classes is of little relevance when comparing between research results, however overlap of class lists between researches is limited, and the width of these sets is varied, e.g. some device type lists include only devices within a certain power range.

There is great variation in device recognition studies, in many factors such as sampling rate, sample resolution, trace size, the number of classes, and the choice of algorithm and features. However, the majority of these researches reports accuracies over 95%, and it is difficult to generate theories pertaining to why each research produced the accuracy which it did. If these accuracies were stressed more, their degree of variation might increase, which in turn might make it feasible to create a qualitative ranking, or at least theories concerning the strengths and weaknesses of various algorithms, features, and sampling scales.

The goal of this research is to investigate the feasibility of electrical device recognition using Crownstones. To start this investigation, the classification method explained in [9] is implemented, due to its extensive use of low frequency data and suitability for low quality, mass recorded data.

3 Research Objectives

In this research, the method of device recognition used in [9] will be tested in a more demanding experiment. A specific measuring device is used for its potential in distribution, and a larger number of classes will be recorded and classified. The intent is to find out whether the accuracy will continue to be high, or may not be sufficient anymore with these factors. Therefore, the main question this research seeks to answer is:

How does the referenced method perform when distinguishing between more classes, based on data collected with the Crownstone?

This is strictly a machine learning problem. The task is to correctly identify the class of each sample within a range of 51 classes, each of which corresponds to a device that has been measured over the course of multiple days. A sample is equal to one day's worth of measurements, from midnight to midnight.

Each device was monitored for at least 30 days, however, it was trivial to collect extra data for some, while others were only in use for a limited time, yielding no measurements for some or most days.

As such devices that are less active overall would pose the same issues in a practical application, their relative deficiency of data was left intact in the interest of experiment realism. As such, there are fewer samples of these devices going into the machine learning tests, which can be verified also by inspecting the confusion matrices in the appendix.

Building and running of the models is done by the Weka toolkit [3]. Each of the classifiers used is included by default in the toolkit, and these are the same classifiers used in [9], which improves the comparability of the results of these experiments and those of the referenced research. Accuracy is found by way of a cross-validation test with 10 folds.

The number of classes distinguished between in this research is larger than in the referenced work, by a margin over 50%. Secondly, as the original 512 features are not documented in detail of implementation, as well as in the interest of simplicity, much fewer features are implemented in this research, a total number of 13. Third, the Crownstones used in measuring have a limited accuracy due to a significant amount of noise, and fourth, the timestamping method may induce slight fluctuations of a second or less. Based on these differences, it is expected that the accuracy found in this research will be lower than the results of the referenced work.

Considering the amount of detail that is lost in taking only noisy low-frequency measurements, it is theorised that the method will not be sufficient for a mass consumer application, and that a steep drop in accuracy will be evident. In this case, it will be clear that more precise and advanced techniques in measuring and classification are needed.

In particular, given the noisiness of the Crownstone, it is expected that it will be very difficult to classify low power devices that will not use enough power to go above the noise margin, and that a finer measuring method will be required to distinguish between such devices.

4 Approach

In order to answer the research question, a new, large dataset is required. For this study, a new, large dataset is acquired from Crownstones over a period of two months, from 51 devices. This data will be stored in an online database. Next, the data will be separated into traces of 24 hours, from midnight till midnight, per day of measurements, per device measured. Features will be extracted from these traces, and used in machine learning tests with the Weka toolkit, according to the method described in [9]. This process will now be explained in more detail.

4.1 Dataset and Acquisition

Measurements are collected by Crownstones individually for all appliances. The Crownstone measures electrical current and tension at a rate of 6 kHz and calculates a mean power usage every 2 seconds, broadcasting this value using Bluetooth Low-Energy, henceforth BLE. Noise accounts for a variation of about 3 W up or down in these measurements.

In each building where measurements are collected (5 in total), one or two Raspberry Pi devices are set up. They run a C++ program to listen to BLE and collect this data, parse it, format it and post it as JSON to an online Couch document database using curl over HTTPS, along with a locally generated timestamp.

Each device has been recorded for at least 30 days. However, several devices are completely inactive for a majority of these days, resulting in recordings that are functionally equivalent to recordings of 'empty' Crownstones with no devices connected. These empty recordings will be labeled as Crownstone recordings instead. "No Device" is a class that is no less important to recognise. This is not only important functionally, but also because the noise of each individual Crownstone may follow unique patterns, and this should not factor too much into the accuracy of identifying the devices plugged into those Crownstones.

4.2 High and Low Power Devices

The Crownstone is not infinitely accurate. The electronic circuit, the presence of radio waves, and the frequency of measurement all contribute to the lossiness and noisiness of the data, and the samples received ultimately are averages taken of six thousand samples per second. Typically, in the data collected in this research, noise levels can go up to +6 Watts and down to -6 Watts. Notably, this research uses an older model of Crownstone, and accuracy is expected to improve with release.

Due to the significant amount of noise produced by the Crownstones, it may turn out difficult to identify whether a low power device is active or inactive, especially when holding all devices to the same standards. Therefore, it may be beneficial to make an initial separation of devices based on their overall levels of power usage. They can be separated into a high power group and a low power group, and each group assigned its own threshold that marks the difference between action and inaction. In this research, this separation is made by visually inspecting a few traces of each device, and identifying which devices have a maximum power usage above 20 W. Those devices are considered high power devices, whereas devices with a maximum power usage equal to or lower than 20 W are considered low power devices.

4.3 Noise Reduction

In their paper, Reinhardt et al. mention using preprocessing to remove noise and outliers from their data. In order to thoroughly explore the accuracies that may be achieved, it was decided that this research would include such preprocessing as well. Simple denoising methods for time series are the moving average and moving median methods, which average a set number of neighbouring points in order to reduce noise. However, this method reduces all detail, and severely lowers peak levels, thereby taking away much important information.

Instead, a Discrete Wavelet Transform, or DWT, was used with a Daubechies level 1 mother wavelet. Using the wavelet transform, the signal is decomposed into high and low frequency subbands, and high frequency subbands with low values are set to zero in an effort to remove uninformative noise and keep important details. For information on wavelet transforms and the Daubechies wavelet, refer to [7]. See Figure 4.3 for a

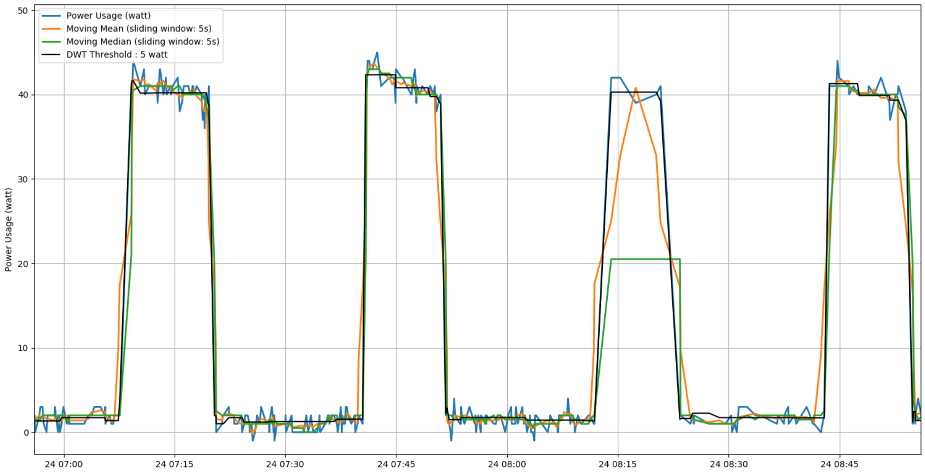


Figure 2: Comparison of the signal in original noisy form and denoised forms.

comparison of a 2 hour trace denoised with each of the methods discussed. It is clear that the DWT better preserves the characteristic peak levels while reducing noise.

The DWT is implemented using a Non-Negative Garrote thresholding function, with a threshold value of 5 W. A garrote thresholding function returns values that lie between those of soft and hard thresholding. Refer to this work [4] for more information on the Non-Negative Garrote thresholding function.

4.4 Feature Extraction

Based on descriptions from [9], the 15 most informative features are implemented and used for classification. However, two of these features are described as ”maximum power level during the complete day” and ”highest power level during activity phases”, and these appear functionally identical, as, by definition, the maximum power level would be observed during an activity phase. For this reason, this is only implemented once. Furthermore, the feature described as ”average power level using current activity phase” is unclear, as no activity phase is ”current” in particular at the time of feature extraction. With two features unable to be used, a total of 13 remain.

We define an ”Activity Phase” as a period in which power usage exceeds a threshold of 3 Watts. When splitting high and low power devices, this threshold becomes 3 Watts for low power devices and 10 Watts for high power devices. The features are then implemented as follows:

1. AverageEnergyPerActivityPhase: For each activity phase. the integral of power usage per second is calculated. Then, the mean of these integrals is returned as

the feature.

2. AveragePeak: Each sample which is preceded and followed by a sample with a lower recorded power usage is considered a ”peak”, and the mean of the power usage of all such peaks is returned as the feature.
3. AveragePower: The mean power usage of all samples is returned as the feature.
4. AveragePowerDuringActivity: Only samples above the activity phase threshold are considered, and the mean power usage of these samples is returned as the feature.
5. DCOffsetOfDiscreteFourier: Originally, this feature was taken from a Discrete Fourier Transform. However, due to the nature of such a transform, the DC offset is equal to the sum of values, so the sum of all power usage values is returned as the feature.
6. EnergyBetween2100And2110: Only samples ranging from 21:00:00 to 21:09:59 are considered, and the integral of power usage per second over this period is returned as the feature.
7. EnergyBetween2130And2140: As above, but for the period from 21:30:00 to 21:39:59.
8. LargestNegativePowerStep: The difference in power usage between each sample and the next is considered, and the largest negative difference is returned as the feature.
9. LargestPositivePowerStep: As above, but the largest positive difference is returned as the feature.
10. LowestPowerDuringActivity: Only samples above the activity phase threshold are considered, and the lowest power usage value among these samples is returned as the feature.
11. MaxPower: The largest power usage value among all samples is returned as the feature.
12. MaxPowerInLastPhase: Only samples within the last activity phase of the 24-hour trace are considered, and the largest power usage value among these samples is returned as the feature.
13. MedianActivityDuration: The duration of each activity phase is considered, and the largest of these durations is returned as the feature.

4.5 Machine Learning

In the interest of consistency and simplicity, the Weka toolkit is used for machine learning, just as it was used in the referenced research. The following classification algorithms are used: Random Tree, Random Forest, Random Committee, Bayesian Network, Naive Bayesian, Bagging, J48, JRip and LogitBoost.

These methods will be applied to the features extracted from the gathered 24 hour traces, in order to find new accuracy statistics that may be compared with the findings of Reinhardt et al. [9].

5 Results and Discussion

In this section, experiments are carried out as planned in previous sections. Data collected from Crownstones is used in machine learning tests using Weka and its built-in classification algorithms, and the results are discussed. The first test focuses on the base performance of the implemented features on the recorded data, the second experiment improves the accuracy with some added complexity, and the third experiment adds a denoising preprocessing step.

5.1 Analysis of Features, Classifiers and Initial Performance

The first experiment of this research investigates the performance of the method using only the 13 implemented features. Preprocessing of the data is limited to the removal of traces that are empty or incomplete. The Random Committee classifier is used, because it was identified as the preferred classifier in [9]. It is worth reiterating that the main differences between this test and the tests in the referred paper are the more contemporary set of devices and the use of Crownstones to gather the data, as well as the use of only 13 of the most relevant features. In this research, cross-validation is performed with 10 folds.

The reported accuracy of this first test is about 84.8%, but when looking more closely at the log, the variance among precision ratings for individual classes is large. Most classes have been predicted with a precision between 75% and 100%, but the Beamer, Electric Rice Cooker, Juicer, and Robert air washer fall far below that range. Some of these classes have particularly limited data, namely about 10 samples each, but overall there is no clear correlation between the number of samples and the accuracy. For an example of the contrary, the Audio Amplifier also yielded only 12 samples, and 11 of them were correctly classified.

It seems impossible to recognise the empty Crownstone class. It is possible that the method is mostly capable of recognising individual Crownstones by their unique noise levels, and therefore incapable of detecting a class recorded from several Crownstones, but it may be more likely that the empty Crownstone offers no unique pattern to recognise over the course of a day, nor even so much as a unique level of constant usage.

Detailed statistics and confusion matrices for all classification experiments can be found in the Appendix.

Furthermore, to gain insight into the value of the 13 features being used and the nature of the collected data, an analysis of information gain per feature is done. Refer to Table 2. The most informative feature is MaxPower. This is unsurprising, as the maximum power usage is strongly defined by the appliance, each requiring an amount of power specific to its purpose and significantly modified by the efficiency of its construction.

The power usage counts around 9 in the evening are also very relevant, as about half of the measured devices are inactive around that time. Typical human behaviour implies that certain activities are usually confined to specific parts of the day, which also supports the idea of turning daily time slots into features.

While Average Peak is listed with a high rank, it should be noted that this will

Score	Feature
2.7827	MaxPower
2.4511	EnergyBetween2130And2140
2.4103	EnergyBetween2100And2110
2.2041	LargestPositivePowerStep
2.1601	LargestNegativePowerStep
2.1271	DCOffsetOfDiscreteFourier
2.0836	AverageEnergyPerActivityPhase
2.0593	AveragePeak
1.9462	AveragePower
1.5164	MedianActivityDuration
1.4034	AveragePowerDuringActivity
0.775	MaxPowerInLastPhase
0.1805	LowestPowerDuringActivity

Table 2: Features ranked by information gain from the noisy data test.

register peaks almost all throughout the noisy data, resulting in a metric related to the average power feature. A quick modified classification test without the Average Peak feature lowers the accuracy to about 82.5%, suggesting that even with such noisy data, the feature may not be entirely trivial.

The Largest Step features are also very useful as they reflect the differences in how quickly devices will enter and exit periods of high power usage, although much of this detail is lost to the limited sampling rate. Still, the capacitors in many devices take much longer than a second to fully charge or discharge, to give an example of a measurable distinction.

LowestPowerDuringActivity comes out least informative. It was expected that this feature would mostly return a value very close above the set activity threshold, the lowest returnable value other than nil, and this does indeed seem to hold for most devices. However, there are a significant number of samples from a few device classes, namely the LCD Monitor (Carolyn's), Induction Cooker, Humidifier, Digital TV Receiver and Desktop Computer, that yield a feature value far above the threshold, implying that these devices transition into and out of activity quickly, without a slope, or at least one slow enough to be measured at this sampling rate.

For the sake of comparison versus the cross-validation test, another experiment was done with a simple split, where 66% of the dataset was used for training, and the remainder for the classification test. This resulted in an accuracy of about 81.9%. Individual accuracies and confusions did not appear to be significantly different when compared to the cross-validation test.

5.2 Low and High Power Devices

Now, more preprocessing is applied for a second test. This time, as planned, the dataset is split into high and low power devices, and different thresholds are used for each in feature extraction. Once again, a cross-validation test is done using the RandomCommittee classifier. Thanks to the manual labour of separating the devices into two categories, which could be automated in a later implementation, accuracy increases drastically to about 94.6%.

All classes that were particularly problematic before are now being predicted with at least 70% precision, save for the Beamer, of which there is very little data, and the empty Crownstone, which is still impossible to recognise. It is not surprising that the problem of classifying the empty Crownstone has remained constant with this change, as the threshold would not have changed, and it was not particularly being confused with high power devices. Without a doubt, there are many households where beamers are used more regularly, and they will likely be easier to recognise with more data.

5.3 Noise Reduction

Finally, noise reduction is applied using the Discrete Wavelet Transform. Another cross-validation test with RandomCommittee classification test is performed on the resulting dataset. Accuracy decreases, curiously, to about 92.6%. The denoising process has reduced the number of traces, and several devices have ended up with too few samples for accurate classification. Despite these outliers, most devices are being classified with a precision above 90%.

Therefore, in all likelihood, denoising will not reduce the accuracy of tests based on larger datasets recorded over several months time. It is also possible that using the same activity threshold for noisy and denoised data is negatively affecting the experiment, and that tuning a new threshold for the denoised data will improve accuracy.

Following up on a theory from the first experiment, another analysis of features by information gain is run. AveragePeak continues to claim relevance, now with a rating of about 2.68, sitting slightly higher in the list, but whether denoising made the feature much more important is questionable. A classification test without the AveragePeak feature even claims a slightly higher accuracy of about 92.9%.

5.4 Classifier Comparison

Next, the other classifiers Random Tree, Random Forest, Bayesian Network, Naive Bayesian, Bagging, J48, JRip and LogitBoost are tested to compare their performance and accuracy. The models were built in 0.86 seconds on a MacBook Pro 15-inch 2017 with a 2,9 GHz Intel Core i7. Refer to Tables 3 and 4.

It is concluded that the Random Forest is the preferred classifier, given the results of this research, for its highest performance in terms of accuracy. It does take proportionally much longer than the Bayesian Network classifier, which has second place for accuracy, but this is insignificant in the current scope. If time becomes more of an issue, the Bayesian Network would be preferred.

Classifier	Noisy	Split	Split & Denoised
RandomForest	85.5397 %	95.7339 %	94.741 %
BayesNet	80.8961 %	95.0832 %	93.5636 %
RandomCommittee	84.888 %	94.6493 %	92.6217 %
LogitBoost	79.3483 %	92.8416 %	91.6797 %
J48	79.3483 %	91.1786 %	89.011 %
Bagging	79.7963 %	90.6725 %	90.1884 %
RandomTree	74.2159 %	90.0217 %	85.8713 %
NaiveBayes	48.0652 %	88.8648 %	87.3626 %
JRip	69.9796 %	82.6464 %	79.5133 %

Table 3: Classifier accuracy comparison. Ranked from highest to lowest split accuracy. Statistics as reported by Weka.

Classifier	Noisy	Split	Split & Denoised
LogitBoost	1.42s	2.31s	0.75s
RandomForest	1.06s	0.78s	0.68s
JRip	0.88s	0.55s	0.28s
Bagging	0.57s	0.1s	0.11s
RandomCommittee	3.35s	0.08s	0.08s
BayesNet	0.02s	0.03s	0.02s
J48	0.06s	0.02s	0.03s
RandomTree	0.02s	0.01s	0.01s
NaiveBayes	0s	0s	0s

Table 4: Classifier build time comparison. Ranked from longest to shortest split build time. Statistics as reported by Weka.

6 Conclusion

Despite expectations, it appears that the classification method from [9] used in this research may be sufficient for commercial use. With limited preprocessing, accuracy can be driven up to an average of about 95%, and most probably this can be improved further with more precise action and a larger dataset to train with. In the scope of a mass sale product that can measure and classify devices in many households over years of time, it is to be expected that the method will be more reliable and precise than in these tests.

However, further analysis of the task of recognising empty Crownstones, as well as the line between the ability to recognise individual Crownstones and the devices plugged into them, is still warranted. Moreover, this method requires a large amount of training data, and cannot be expected to recognise devices before they have been plugged in for a full 24-hour day and used at least once in that time.

7 Future Work

The considerations, insights and results of this research invite much further research. Feature extraction, trace selection and denoising may be further improved. It is, for example, expected that the split between low and high power feature extraction can be automated so that the activity threshold is dynamically calculated for each device class.

The high accuracy achieved invites further stress upon the method, which would be introduced by tests involving even larger numbers of classes. Moreover, the sheer effectiveness of a method this easy to implement urges one to drive up the requirements. With the potential of mass distribution of the Crownstone, the promise of improved accuracy in newer models, and the ability to extract low and high frequency data from measured devices, it would be desirable to develop the ability to recognise devices much faster, as well as their specific activities and electrical decay.

Both the notion of faster recognition and that of a tougher challenge may be followed by way of tests with smaller traces: especially with some tweaks to feature extraction, it may be possible to use this method on traces of 1 hour or less, given that the device is active during this trace. It may also be interesting to perform tests using data with a lower frequency, which could be emulated by taking the dataset from this research and removing a portion of samples.

Appendices

A Weka Output

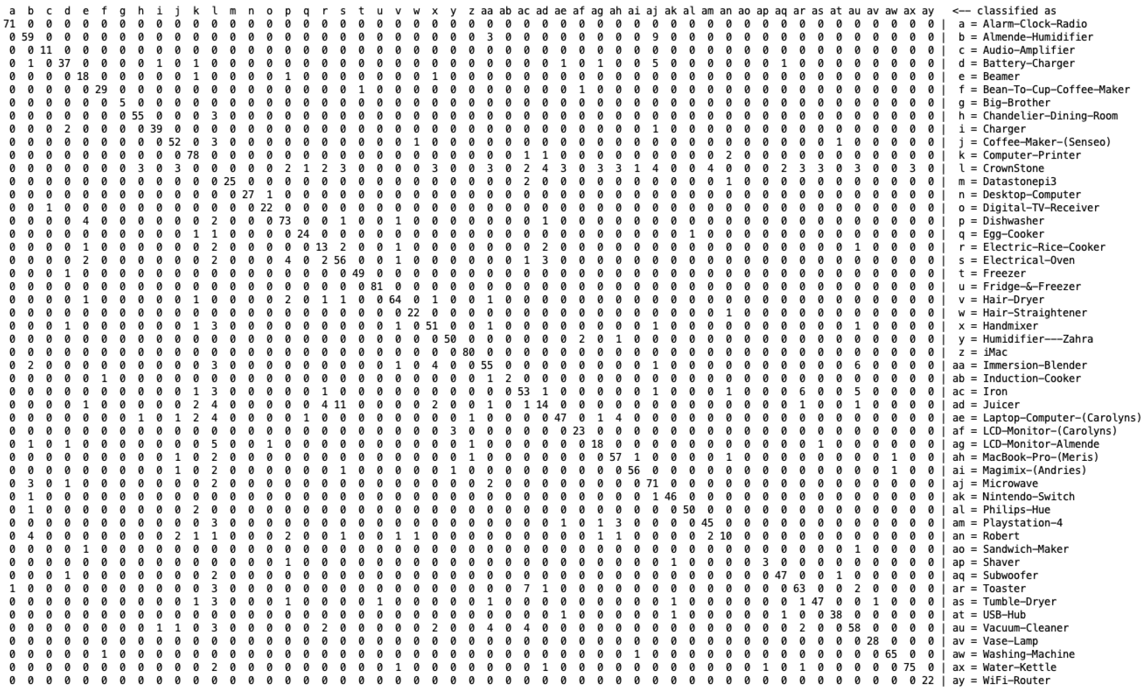
This Appendix section shares relevant data obtained from the Weka toolkit when performing the experiments of this research. Each test is accompanied by a table that lists accuracy statistics and the total number of traces, and by a confusion matrix that elaborates on the classification performance for each device class individually.

A.1 Test 1: Noisy Data

Displayed here are the most relevant statistics as provided by the Weka toolkit concerning the test that is observed in section 5.1.

Correctly Classified Instances	2084	84.888 %
Incorrectly Classified Instances	371	15.112 %
Kappa statistic	0.8451	
Mean absolute error	0.0099	
Root mean squared error	0.0722	
Relative absolute error	25.8747 %	
Root relative squared error	52.2004 %	
Total Number of Instances	2455	

Table 5: Accuracy Statistics for Classification of Noisy Data



```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u  v  w  x  y  z aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay   <-- classified as
 71  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  a = Alarm-Clock-Radio
  0 59  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  b = Almende-Humidifier
  0  0 11  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  c = Audio-Amplifier
  0  1  0 37  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  5  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0 |  d = Battery-Charger
  0  0  0  0 18  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  e = Beamer
  0  0  0  0  0 29  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  f = Bean-To-Cup-Coffee-Maker
  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  g = Big-Brother
  0  0  0  0  0  0  0 55  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  h = Chandelier-Dining-Room
  0  0  0  2  0  0  0  0 39  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  i = Charger
  0  0  0  0  0  0  0  0  0 52  0  3  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0 |  j = Coffee-Maker-(Senseo)
  0  0  0  0  0  0  0  0  0  0 78  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0 |  k = Computer-Printer
  0  0  0  0  0  0  0  3  0  3  0  0  0  0  0  2  1  2  3  0  0  0  0  3  0  0  3  0  2  4  3  0  3  3  1  4  0  0  4  0  0  0  2  3  3  0  3  0  0  3  0 |  l = CrownStone
  0  0  0  0  0  0  0  0  0  0  0  0 25  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 |  m = Datastonepi3
  0  0  0  0  0  0  0  0  0  0  0  0  0 27  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  n = Desktop-Computer
  0  0  1  0  0  0  0  0  0  0  0  0  0  0 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  o = Digital-TV-Receiver
  0  0  0  0  4  0  0  0  0  0  0  2  0  0  0 73  0  0  1  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  p = Dishwasher
  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0 24  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 |  q = Egg-Cooker
  0  0  0  0  1  0  0  0  0  0  0  2  0  0  0  0  0 13  2  0  0  1  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 |  r = Electric-Rice-Cooker
  0  0  0  0  2  0  0  0  0  0  0  2  0  0  0  4  0  2 56  0  0  1  0  0  0  0  0  0  1  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  s = Electrical-Oven
  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 49  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  t = Freezer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 81  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  u = Fridge-&-Freezer
  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  2  0  1  1  0  0 64  0  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  v = Hair-Dryer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 |  w = Hair-Straightener
  0  0  0  1  0  0  0  0  0  0  1  3  0  0  0  0  0  0  0  0  0  1  0 51  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 |  x = Handmixer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 50  0  0  0  0  0  0  2  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  y = Humidifier---Zahra
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 80  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  z = iMac
  0  2  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  1  0  4  0  0 55  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  6  0  0  0  0 | aa = Immersion-Blender
  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ab = Induction-Cooker
  0  0  0  0  0  0  0  0  0  0  1  3  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0 53  1  0  0  0  0  0  1  0  0  0  1  0  0  0  6  0  0  5  0  0  0  0 | ac = Iron
  0  0  0  0  1  0  0  0  0  0  2  4  0  0  0  0  0  4 11  0  0  0  0  2  0  0  1  0  1 14  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  0 | ad = Juicer
  0  0  0  0  0  0  0  1  0  1  2  4  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0 47  0  1  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ae = Laptop-Computer-(Carolyns)
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0 23  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | af = LCD-Monitor-(Carolyns)
  0  1  0  1  0  0  0  0  0  0  0  5  0  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0 18  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0 | ag = LCD-Monitor-Almende
  0  0  0  0  0  0  0  0  0  1  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0 57  1  0  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0 | ah = MacBook-Pro-(Meris)
  0  0  0  0  0  0  0  0  0  1  0  2  0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0 56  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0 | ai = Magimix-(Andries)
  0  3  0  1  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0 71  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | aj = Microwave
  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 46  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ak = Nintendo-Switch
  0  1  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 50  0  0  0  0  0  0  0  0  0  0  0  0  0 | al = Philips-Hue
  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  3  0  0  0  0 45  0  0  0  0  0  0  0  0  0  0  0  0 | am = Playstation-4
  0  4  0  0  0  0  0  0  0  2  1  1  0  0  0  2  0  0  1  0  0  1  1  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  2 10  0  0  0  0  0  0  0  0  0  0  0 | an = Robert
  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 | ao = Sandwich-Maker
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  3  0  0  0  0  0  0  0  0  0 | ap = Shaver
  0  0  0  1  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 47  0  0  1  0  0  0  0  0 | aq = Subwoofer
  1  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  1  0  0  0  0  0  0  0  0  0  0  0  0  0 63  0  0  2  0  0  0  0 | ar = Toaster
  0  0  0  0  0  0  0  0  0  0  1  3  0  0  0  1  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1 47  0  0  1  0  0  0 | as = Tumble-Dryer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0  1  0  0 38  0  0  0  0  0 | at = USB-Hub
  0  0  0  0  0  0  0  0  1  1  0  3  0  0  0  0  0  2  0  0  0  0  0  2  0  0  4  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0 58  0  0  0  0 | au = Vacuum-Cleaner
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 28  0  0  0 | av = Vase-Lamp
  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 65  0  0 | aw = Washing-Machine
  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0 75  0 | ax = Water-Kettle
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 22 | ay = WiFi-Router
```

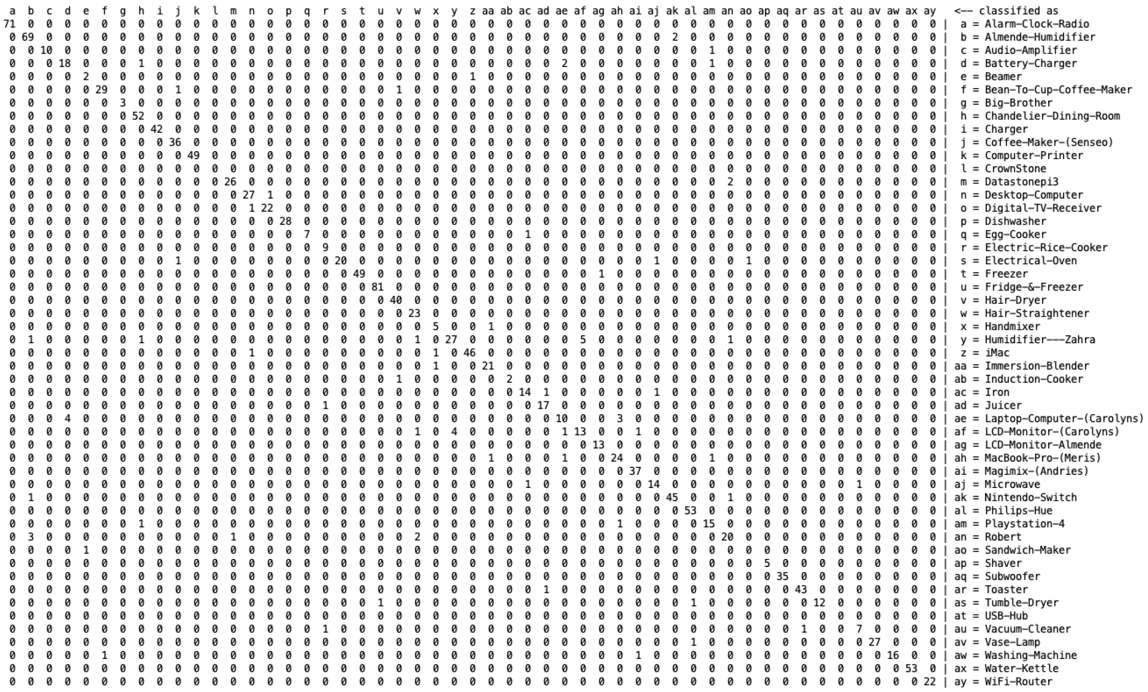
Figure 3: Confusion Matrix for Classification of Noisy Data

A.2 Test 2: Split Data

Displayed here are the most relevant statistics as provided by the Weka toolkit concerning the test that is observed in section 5.2.

Correctly Classified Instances	1309	94.6493 %
Incorrectly Classified Instances	74	5.3507 %
Kappa statistic	0.9449	
Mean absolute error	0.0045	
Root mean squared error	0.0425	
Relative absolute error	11.8255 %	
Root relative squared error	30.8403 %	
Total Number of Instances	1383	

Table 6: Accuracy Statistics for Classification of Split Data



```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u  v  w  x  y  z aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay   <-- classified as
 71  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  a = Alarm-Clock-Radio
  0 69  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  b = Almende-Humidifier
  0  0 10  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0 |  c = Audio-Amplifier
  0  0  0 18  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0 |  d = Battery-Charger
  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  e = Beamer
  0  0  0  0  0 29  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  f = Bean-To-Cup-Coffee-Maker
  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  g = Big-Brother
  0  0  0  0  0  0  0 52  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  h = Chandelier-Dining-Room
  0  0  0  0  0  0  0  0 42  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  i = Charger
  0  0  0  0  0  0  0  0  0 36  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  j = Coffee-Maker-(Senseo)
  0  0  0  0  0  0  0  0  0  0 49  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  k = Computer-Printer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  l = CrownStone
  0  0  0  0  0  0  0  0  0  0  0  0 26  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0 |  m = Datastonepi3
  0  0  0  0  0  0  0  0  0  0  0  0  0 27  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  n = Desktop-Computer
  0  0  0  0  0  0  0  0  0  0  0  0  0  1 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  o = Digital-TV-Receiver
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  p = Dishwasher
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  q = Egg-Cooker
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  r = Electric-Rice-Cooker
  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0 20  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0 |  s = Electrical-Oven
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 49  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  t = Freezer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 81  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  u = Fridge-&-Freezer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  v = Hair-Dryer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  w = Hair-Straightener
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  x = Handmixer
  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0 27  0  0  0  0  0  0  5  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 |  y = Humidifier---Zahra
  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0 46  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  z = iMac
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0 21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | aa = Immersion-Blender
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ab = Induction-Cooker
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 14  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ac = Iron
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ad = Juicer
  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ae = Laptop-Computer-(Carolyns)
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  4  0  0  0  0  0  1 13  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | af = LCD-Monitor-(Carolyns)
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 13  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ag = LCD-Monitor-Almende
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  0  0 24  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0 | ah = MacBook-Pro-(Meris)
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 37  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ai = Magimix-(Andries)
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0 14  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 | aj = Microwave
  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 45  0  0  1  0  0  0  0  0  0  0  0  0  0  0 | ak = Nintendo-Switch
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 53  0  0  0  0  0  0  0  0  0  0  0  0  0 | al = Philips-Hue
  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 15  0  0  0  0  0  0  0  0  0  0  0  0 | am = Playstation-4
  0  3  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 20  0  0  0  0  0  0  0  0  0  0  0 | an = Robert
  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ao = Sandwich-Maker
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0 | ap = Shaver
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 35  0  0  0  0  0  0  0  0 | aq = Subwoofer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 43  0  0  0  0  0  0  0 | ar = Toaster
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0 12  0  0  0  0  0  0 | as = Tumble-Dryer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | at = USB-Hub
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  7  0  0  0  0 | au = Vacuum-Cleaner
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0 27  0  0  0 | av = Vase-Lamp
  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 16  0  0 | aw = Washing-Machine
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 53  0 | ax = Water-Kettle
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 22 | ay = WiFi-Router
```

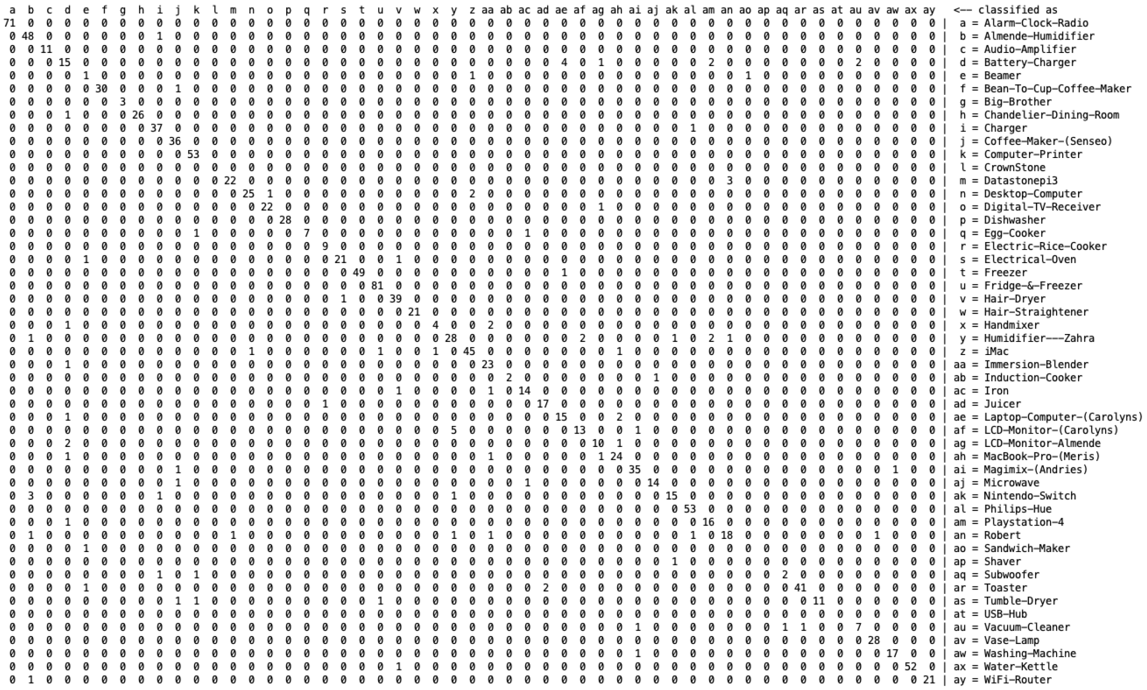
Figure 4: Confusion Matrix for Classification of Split Data

A.3 Test 3: Split and Denoised Data

Displayed here are the most relevant statistics as provided by the Weka toolkit concerning the test that is observed in section 5.3.

Correctly Classified Instances	1180	92.6217 %
Incorrectly Classified Instances	94	7.3783 %
Kappa statistic	0.9239	
Mean absolute error	0.0056	
Root mean squared error	0.0486	
Relative absolute error	14.6633 %	
Root relative squared error	35.2208 %	
Total Number of Instances	1274	

Table 7: Accuracy Statistics for Classification of Split and Denoised Data



```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u  v  w  x  y  z aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay   <-- classified as
 71  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  a = Alarm-Clock-Radio
  0 48  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  b = Almende-Humidifier
  0  0 11  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  c = Audio-Amplifier
  0  0  0 15  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  1  0  0  0  0  0  2  0  0  0  0  0  0  0  2  0  0  0  0 |  d = Battery-Charger
  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0 |  e = Beamer
  0  0  0  0  0 30  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  f = Bean-To-Cup-Coffee-Maker
  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  g = Big-Brother
  0  0  0  1  0  0  0 26  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  h = Chandelier-Dining-Room
  0  0  0  0  0  0  0  0 37  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 |  i = Charger
  0  0  0  0  0  0  0  0  0 36  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  j = Coffee-Maker-(Senseo)
  0  0  0  0  0  0  0  0  0  0 53  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  k = Computer-Printer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  l = CrownStone
  0  0  0  0  0  0  0  0  0  0  0  0 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0  0 |  m = Datastonepi3
  0  0  0  0  0  0  0  0  0  0  0  0  0 25  1  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  n = Desktop-Computer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0 22  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  o = Digital-TV-Receiver
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  p = Dishwasher
  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  7  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  q = Egg-Cooker
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  9  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  r = Electric-Rice-Cooker
  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 21  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  s = Electrical-Oven
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 49  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  t = Freezer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 81  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  u = Fridge-&-Freezer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0 39  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  v = Hair-Dryer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 21  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  w = Hair-Straightener
  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  x = Handmixer
  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  2  0  0  0  0  1  0  2  1  0  0  0  0  0  0  0  0  0  0  0 |  y = Humidifier---Zahra
  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1  0  0  1  0 45  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  z = iMac
  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 23  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | aa = Immersion-Blender
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ab = Induction-Cooker
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1  0 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ac = Iron
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ad = Juicer
  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 15  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ae = Laptop-Computer-(Carolyns)
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  0  0  0  0  0  0 13  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | af = LCD-Monitor-(Carolyns)
  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 10  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ag = LCD-Monitor-Almende
  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1 24  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ah = MacBook-Pro-(Meris)
  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 35  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0 | ai = Magimix-(Andries)
  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0 14  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | aj = Microwave
  0  3  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 15  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ak = Nintendo-Switch
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 53  0  0  0  0  0  0  0  0  0  0  0  0  0 | al = Philips-Hue
  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 16  0  0  0  0  0  0  0  0  0  0  0  0 | am = Playstation-4
  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  1 18  0  0  0  0  0  0  0  1  0  0  0 | an = Robert
  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ao = Sandwich-Maker
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | ap = Shaver
  0  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0 | aq = Subwoofer
  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0 41  0  0  0  0  0  0  0 | ar = Toaster
  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 11  0  0  0  0  0  0 | as = Tumble-Dryer
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 | at = USB-Hub
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  1  1  0  0  7  0  0  0  0 | au = Vacuum-Cleaner
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 28  0  0  0 | av = Vase-Lamp
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0 | aw = Washing-Machine
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 52  0 | ax = Water-Kettle
  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 21 | ay = WiFi-Router
```

Figure 5: Confusion Matrix for Classification of Split and Denoised Data

References

[1] Crownstone. Crownstone. `https://crownstone.rocks/`.

[2] Frank Englert, Till Schmitt, Sebastian Kößler, Andreas Reinhardt, and Ralf Steinmetz. How to auto-configure your smart home?: High-resolution power measurements to the rescue. In *Proceedings of the fourth international conference on Future energy systems*, pages 215–224. ACM, 2013.

[3] Eibe Frank, Mark Hall, Peter Reutemann, and Len Trigg. Weka. `https://www.cs.waikato.ac.nz/ml/weka/`.

[4] Hong-Ye Gao. Wavelet shrinkage denoising using the non-negative garrote. *Journal of Computational and Graphical Statistics*, 7(4):469–488, 1998.

[5] Takekazu Kato, Hyun Sang Cho, Dongwook Lee, Tetsuo Toyomura, and Tatsuya Yamazaki. Appliance recognition from electric current signals for information-energy integrated network in home environments. In *Ambient Assistive Health and Wellness Management in the Heart of the City*, pages 150–157. Springer, 2009.

[6] HY Lam, GSK Fung, and WK Lee. A novel method to construct taxonomy electrical appliances based on load signatures. *Consumer Electronics, IEEE Transactions on*, 53(2):653–660, 2007.

[7] Tao Li, Qi Li, Shenghuo Zhu, and Mitsunori Ogihara. A survey on wavelet applications in data mining. SIGKDD Explorations, 2002.

[8] Francesca Paradiso, Federica Paganelli, Antonio Luchetta, Dino Giuli, and Pino Castrogiovanni. Ann-based appliance recognition from low-frequency energy monitoring data. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–6. IEEE, 2013.

[9] Andreas Reinhardt, Paul Baumann, Daniel Burgstahler, Matthias Hollick, Hristo Chonov, Marc Werner, and Ralf Steinmetz. On the accuracy of appliance identification based on distributed load metering data. In *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*, pages 1–9. IEEE, 2012.

[10] Andreas Reinhardt, Dominic Burkhardt, Manzil Zaheer, and Ralf Steinmetz. Electric appliance classification based on distributed high resolution current sensing. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, pages 999–1005. IEEE, 2012.

[11] Antonio Ridi, Christophe Gisler, and Jean Hennebert. Automatic identification of electrical appliances using smart plugs. In *Systems, Signal Processing and their Applications (WoSSPA), 2013 8th International Workshop on*, pages 301–305. IEEE, 2013.

[12] Antonio Ridi, Christophe Gisler, and Jean Hennebert. A survey on intrusive load monitoring for appliance recognition. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 3702–3707. IEEE, 2014.

[13] Damien Zufferey, Christophe Gisler, Omar Abou Khaled, and Jean Hennebert. Machine learning approaches for electric appliance classification. In *Information*

Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on, pages 740–745. IEEE, 2012.