Classification of dwelling characteristics with machine learning algorithms based on smart meter & weather data

Teo Curcic[1], *Rajeev Robin* Kalloe[1], *Merel Alexandra* Kreszner[2], *Olivier* van Luijk[1], *Santiago* Puertas Puchol[3], *Emilio* Caba Batuecas[3] *& Tadeo Baldiri* Salcedo Rahola[1]

1 Faculty of Technology, Innovation and Society, The Hague University of Applied Sciences, Johanna Westerdijkplein 75, 2521 EN The Hague, The Netherlands
2 Faculty of Engineering & ICT, Avans University, Lovensdijkstraat 61 - 63, 4818 AJ Breda, the Netherlands
3 Escuela Politécnica Superior, University Francisco de Vitoria, Carretera Pozuelo a Majadahonda, Pozuelo de Alarcón, Madrid, Spain

Abstract - Machine learning models have proven to be reliable methods in classification tasks. However, little research has been done on classifying dwelling characteristics based on smart meter & weather data before. Gaining insights into dwelling characteristics can be helpful to create/improve the policies for creating new dwellings at NZEB standard. This paper compares the different machine learning algorithms and the methods used to correctly implement the models. These methods include the data pre-processing, model validation and evaluation. Smart meter data was provided by Groene Mient, which was used to train several machine learning algorithms. The models that were generated by the algorithms were compared on their performance. The results showed that Recurrent Neural Network (RNN) performed the best with 96% of accuracy. Cross Validation was used to validate the models, where 80% of the data was used for training purposes and 20% was used for testing purposes. Evaluation metrices were used to produce classification reports, which can indicate which of the models work the best for this specific problem. The models were programmed in Python.

1 Introduction

In the last few years, the European Commission has set various ambitious targets in order to reduce CO2 emissions. As a result of this, the Dutch government has been regulating the energy sector to move towards the use of green energy in order to avoid the use of natural gasses completely by the end of 2050 [1]. Moreover, in the Netherlands there has been a growing concern with the use of natural gas, because of the exponential growth of earthquakes caused by the extraction of gas from the Groningen gas field. Accordingly, the urgency of reducing the use of natural gasses in dwellings is growing strongly. Although most energy in the Netherlands is used by industry, the build environment is responsible for 28% of the energy use from which 71% is for heating purposes. Currently, 87% of the buildings heating energy needs are covered with the use of natural gas [2]. The technologies that currently exist which could help us reduce the use of natural gasses, are easier to implement in the build environment rather than the industry, since the industry often has higher temperature needs than the build environment [3].

To reduce the use of gas in the built environment, some new and renovated dwellings have already been constructed to Net-Zero Energy Building (NZEB) standard. This means that for these dwellings the total amount of energy used on an annual basis is equal to the amount of renewable energy created on site. The type of energy produced and used by this type of dwellings in the Netherlands is only electrical [4]. The electricity energy use and production patterns of NZEB dwellings differs a lot from the average dwellings, because of the variety of heating and energy producing systems used.

Since these dwellings have such different energy patterns, new approaches are needed to be able to take technical/political decisions about what type of technologies to promote. Government-organisations can therefore use this information to create/improve (existing) policies. To set up these policies, information about dwelling energy usage and delivery can be collected from the dwellings. A good technique to collect that information is via smart meters.

By 2020, 80% of the households in The Netherlands will have a smart meter. A smart meter measures the amount of energy taken from and sent to the net [5, 6]. This data, in combination with publicly available data, could allow determining dwelling characteristics and the energy use behaviour of its residents. This information could be used to optimize the current dwellings energy systems, gain more knowledge about energy use patterns and to propose possible improvements with the goal to reduce CO2 emissions related to energy use.

Previous research shows insights about household electricity consumption and CO2 emissions on dwellings in the Netherlands. It explores the effect of smart meter, appliance efficiency and consumer behaviour on reducing electricity consumption in the Netherlands. The results show their effect on electricity consumption and suggest that further effort is required to control and reduce it. Insights from the paper suggest that future studies should disaggregate with respect to several factors in electricity consumption as George Papachristos has stated in his paper [7].

Based on the research of Papachristos, behavioural patterns have been formed based on the characteristics of dwellings and electricity consumption. It analyses the appliance uses in the Dutch housing stock and define behavioural patterns and profiles of electricity consumption in detail. This has been done with survey data which were collected from 323 dwellings in the Netherlands on appliance ownership and use of electricity [8].

Currently, smart meters are being installed in most of the Dutch dwellings. Data from the smart meters contains important information about the minimum and maximum amount of energy circling back and forward in the net. This is needed to change the current net infrastructure, because the current infrastructure is not built for receiving as much energy from dwellings. Energy produced at dwellings should be able to be sent back to the net without little to no energy loss. This energy can then be used at moments without sun and/or wind [9].

Thorough analysis of 15-minute residential smart meter datasets, it is possible to identify possible value propositions of smart meter measurements. The results showed that for different applications, the communication needs from meters to control-centers, data storage capabilities, and the complexity of data processing intelligence varies significantly [10].

When introducing the people in the dwelling with their consumer behaviour, this can be used to change their behaviour and therefore reduce the yearly energy cost. The consumption behaviour is based on the amount of energy every device in the dwelling is using, visualized by logistic regression machine learning [11]. Another research proves that the use of machine learning algorithms can be used to forecast residential gas consumption based on energy consumption data and weather data. Gaining insight into the energy consumption with machine learning algorithms can be helpful in balancing the grid and insights in how to reduce the energy consumption can be received [12, 13].

This research focuses on gaining insights on the dwelling characteristics by using machine learning algorithms on smart meter data in combination of weather data. Specifically, the paper compares the accuracy of the models classifying the dwellings by type of heating system, number of solar panels and number of inhabitants.

2 Methodology

Supervised machine learning and neural network methods have been used in this research. Machine learning has proven to be a solution to different problems, including when it comes to analysing smart meter data [10, 12, 14]. Logistic Regression, Support Vector Machine and K-Nearest Neighbours are one of the most used machine learning algorithms today. These algorithms are hardly been used to classify smart meter data; however, they are commonly used to work efficient with the classification of sequential data [15]. In addition to supervised machine learning, neural networks show very promising results when predicting/classifying data. Recurrent Neural Networks are a type of neural networks, which are designed to detect and recognize patterns in sequences of data. All these algorithms are described in their respective subsections.

Furthermore, the data collecting & pre-processing methods used are described as well as the classification metrics to evaluate the effectiveness and performance of the different machine learning models. This section also specifies the used environmental settings for all the models. All the analysis and modelling were done in Python.

2.1 Data collecting and pre-processing

The original dataset was obtained from the processing of an Excel-file of two sheets. This file contains records in the form of time series of energy delivery and consumption data from 33 dwellings of a neighbourhood called Groene Mient in the Netherlands. The data was extracted from the smart meters installed in these dwellings. The smart meters collected energy delivery and consumption records every 15 minutes from 11-07-2017 till 31-05-2019.

In the data cleaning phase, the Python-scripts pre-processed the rows containing missing records and accumulated consumption data so that correct learning models could be created in the next phase. The dataset consists of different data depending on each dwelling, because the smart meters were installed at different times. Therefore, the dataset was reduced for the dwellings which had less data compared to the other dwellings. This process took out 5 dwellings from the dataset, which means that every machine learning model was built based on the 28 dwellings that were left in the dataset.

Apart from the smart meters data, weather data has been gathered from the *Koninklijk Nederlands Meteorologisch Instituut (KNMI)* [13]. This dataset helps us to build the models providing more data about the environment which influence directly to the energy delivery and consumption. The data consists of the temperature, duration of sunshine, global radiation and cloud cover indexes. Moreover, dummy variables, variables that are created from values within the dataset, have been created based on the timestamps, including are the following: hour of the day, day of the week, day of the month, week of the month, month and season.

In order to let the algorithms find the underlying patterns to correctly predict the targets, it is necessary to perform a transformation of the original dataset by grouping the energy delivery and consumption into series of daily and weekly data. Depending on which aggregated data an algorithm is trained with, it always predicted better with aggregated data instead of data per 15 minutes. For the weather data, the mean has been computed of every variable during the week, and for the dummy variables, the mode has been calculated. In the case of the Recurrent Neural Network, this grouping was not necessary to do.

2.2 Implementation of Machine Learning

Machine learning is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment. They are considered the working horse in the new era of the so-called big data. Machine learning can be divided according to the nature of the data labelling into supervised, unsupervised, and semi-supervised. Supervised learning is used to estimate an unknown (input, output) mapping from known (input, output) samples, where the output is labelled (e.g. classification and regression). In unsupervised learning, only input samples are given to the learning system (e.g. clustering and estimation of probability density function) [16].

The focus was on supervised learning, because it allows us to classify different targets, including which type of heating system, number of solar panels and number of inhabitants is the most suitable for a certain dwelling by using the available datasets. This is done by using classification algorithms, which classify data into two or more classes.

The following machine learning algorithms have been used in order to classify based on the data:

- Logistic Regression
- Support Vector Machine
- K-Nearest Neighbours
- Recurrent Neural Network

2.2.1 Logistic Regression

Logistic Regression is a supervised machine learning algorithm which can be used to solve classifications problems. Logistic Regression uses a linear equation to calculate a value. The calculated value can be anywhere between negative infinity and positive infinity. The output needs to be between 0 and 1 to make the classification. To scale the output to a value between 0 and 1, the sigmoid function is used: $x = \frac{1}{1+\mathrm{e}^{-z}}$ where ‘z’ is the output value (figure 1).

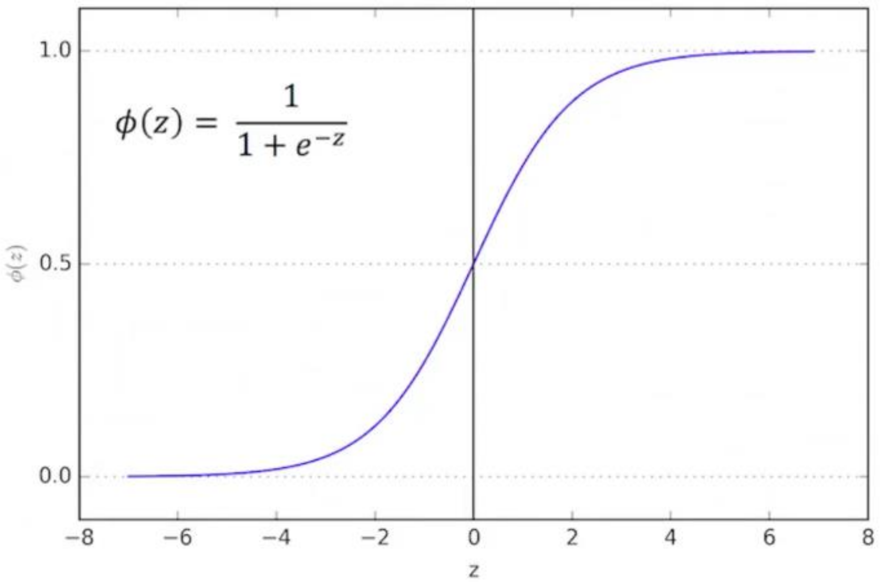


Figure 1: The sigmoid function

When using Logistic Regression, a threshold is specified that indicates at which value the output will be put into one class vs. the other class. A threshold with the value 0.5 means that a class with a 50% (or greater) probability will be classified as class2 and a class with a probability less than 50% as class1. Logistic Regression can be divided in binary (where the model classifies the data in two classes) and multi-class classification (where the model classifies the data in two or more classes). Depending on the data, multi-class classification was used since the prediction for each label could be divided in 2 or more classes.

The model is fed with 9744 rows of aggregated data per day which contains delivery, consumption and KNMI data. Logistic Regression was used to classify what kind of heating system is used, how many solar panels are installed and how many inhabitants are in this dwelling. For every prediction a different model is used. To improve the accuracy of the model, the hyperparameter ‘C’ can be changed. For small values of C, the regularization strength is increased which will create simple models which underfit the data. For big values of C vice versa.

2.2.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning model with associated learning algorithms that analyse data used for classification and regression analysis. SVM is highly preferred by many as it produces significant accuracy with less computation power. Although it can be used for both classification and regression analysis, SVM is widely used for classification objectives.

The objective of SVM is to find a hyperplane in an N-dimensional space (N = number of features) that classifies the datapoints. Hyperplanes are decision boundaries that help classify data points. Data points falling on one side of the hyperplane can be attributed as a certain class. The dimension of the hyperplane depends on the number of features. SVM can be divided in binary (where the model classifies the data into two classes) and multi-class classification (where the model classifies the data into two or more classes). For the chosen solution, multi-class classification was used since the type of heating system can be divided in three classes.

The model was fed with 9744 rows of aggregated data per day which contains delivery, consumption and KNMI data. SVM was used to classify the type of heating system. The model was configured by specifying the 'gamma' parameter, which defines how far the influence of a single training example reaches. This parameter has been set to 'auto' so that the model itself can determine the most efficient value for 'gamma'.

2.2.3 K-Nearest Neighbours

K-Nearest Neighbours (KNN) can be used for both classification and regression predictive problems and is a supervised machine learning algorithm. The goal of the algorithm is to learn a function $h: X \rightarrow Y$ so that given an unseen observation x, h(x) can confidently predict the corresponding output y.

In the classification setting, KNN essentially boils down to forming a majority vote between the K most similar instances to a given "unseen" observation. Similarity is defined according to a distance metric between two data points. The KNN classifier is commonly based on the Euclidean distance between a test sample and the specified training samples. Let x_i be an input sample with p features $(x_{i1}, x_{i2}, ..., x_{ip})$, n be the total number of input samples ($i=1, 2, ..., n$). The Euclidean distance between sample x_i and x_j is defined as:

$$d(x_i,\ x_j\ ) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + \ (x_{ip} - x_{jp})^2}$$

The model is fed with 2912 rows of aggregated data per week which contains delivery, consumption and KNMI data. KNN was used to classify what kind of heating system is used, how many solar panels are installed and how many inhabitants are in this dwelling. For every prediction, a different model is used. To improve the accuracy of the model, the hyperparameter 'n_neighbors', which determines how many neighbours have to be checked to set the class for the new sample; and 'p', which represents the power parameter for the Minkowski metric can be changed.

2.2.4 Recurrent Neural Network

Recurrent Neural Networks (RNN) are one type of neural networks which are designed to detect and recognize patterns in sequences of data. The dataset from Groene Mient is suitable for RNN, because it contains time and sequential data. That means that the dataset contains temporal dimensions, which is useful for RNN's. Furthermore, the dataset contains different sizes of sequences with intervals of 15 minutes. The powerful tool of RNN is that they have a certain type of memory, and that is also part of the human brain condition. As they have feedforward connections, they can use the outputs as part of the next input moment after moment [17].

2.2.4.1 Long short-term memory

Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture that has shown to outperform traditional RNNs on numerous temporal processing task [17].

LSTM is a variant of the RNN maintaining the error more constantly, therefore, they are used to long time series data memorization in order to continue learning over many more time steps than the traditional RNN. One relevant point of the LSTM is that they have a forget gate; using this feature the network can forget some low-quality patterns and start over others. Basically, LSTM performs better than other recurrent neural networks when the goal is to learn from very long-term data sequences. The ability to forget, remember and update the information make better adjustments one step ahead of RNNs. The input data shape of the LSTM Keras must be three dimensional. The first dimension measures the batch size, the second one measure the time-steps and the last one dimension measures the number of units in one input sequence [18].

The network is fed with 8925 samples with 92 data records (96 should be the optimum, because there are 92 data records per day, but the entire dataset has no possibility to divide it by 96, so the closest number is 92) and 10 units (each data record has 10 features to train delivery, consumption, KNMI data and dummy variables).

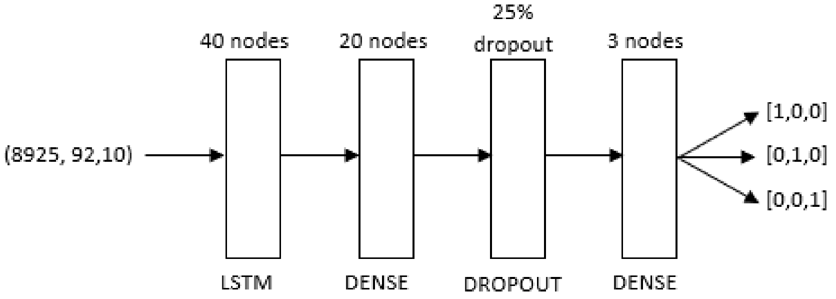


Figure 2: LSTM heating system architecture

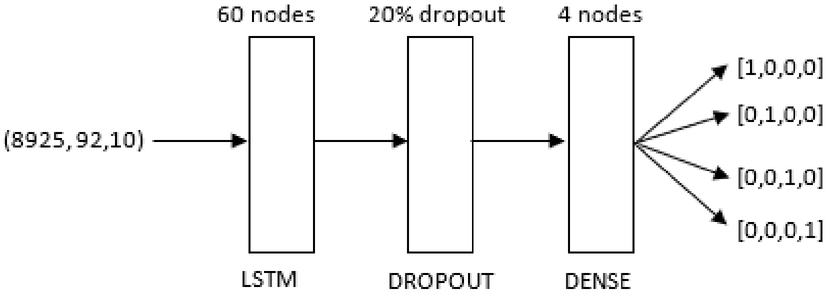


Figure 3: LSTM number of inhabitants architecture

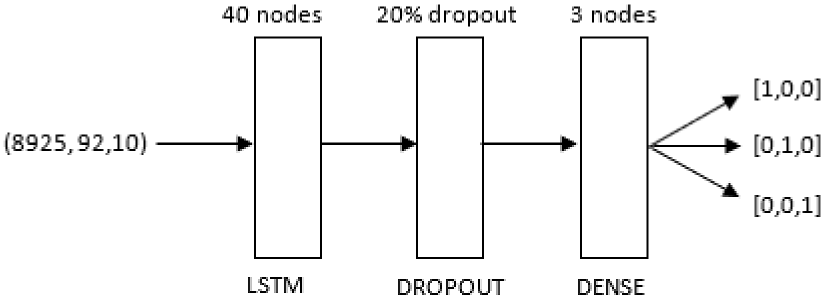


Figure 4: LSTM number of solar panels architecture

The architecture of the model (figure 2, 3, 4) is composed of 4 layers, the input layer is the LSTM with a dropout of 20%, a recurrent dropout of 10% and 40 nodes. The second layer is a Dense layer (simple neural net, the input neuron is connected to the output neuron) of 20 nodes using a RELU activation function. The third layer is a dropout of 25% (randomly selected neurons are ignored during training. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass) [19]. Finally, the output is another Dense layer using a “softmax” activation function composed of 3 nodes (depends on the number of classes to predict). Dropouts are used in order to prevent overfitting, forcing the model to learn pattern in different ways.

The rectified linear activation function (RELU) is a piecewise linear function that will output the input directly if is positive, otherwise, it will output zero (figure 5).

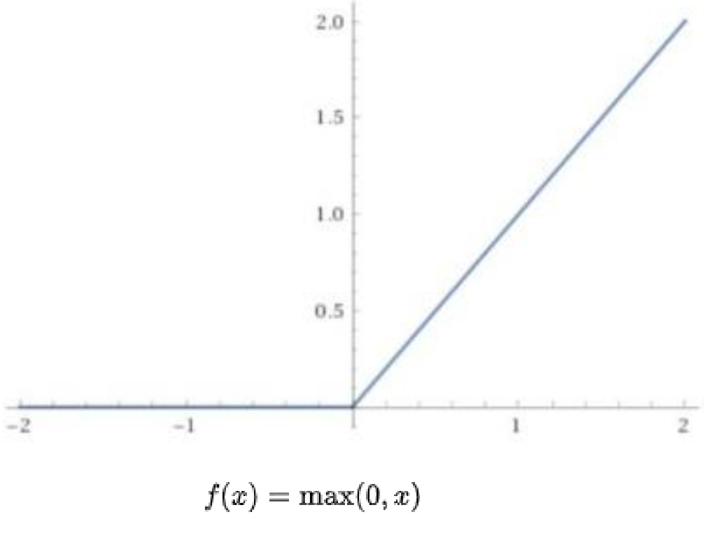
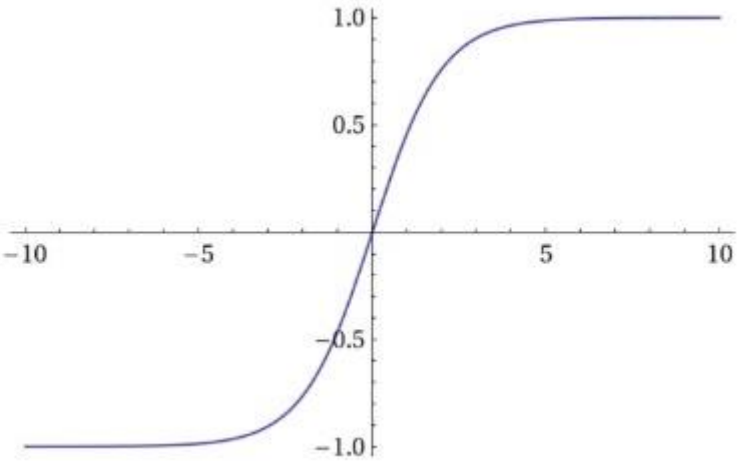


Figure 5: The RELU activation function

By assigning a softmax activation function (figure 6), a generalization of the logistic function, on the output layer of the neural network (or a softmax component in a component-based network) for categorical target variables, the outputs can be interpreted as posterior probabilities. This is useful in classification as it gives a certainty measure on classifications.



$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{c} e^{x_j}}$$

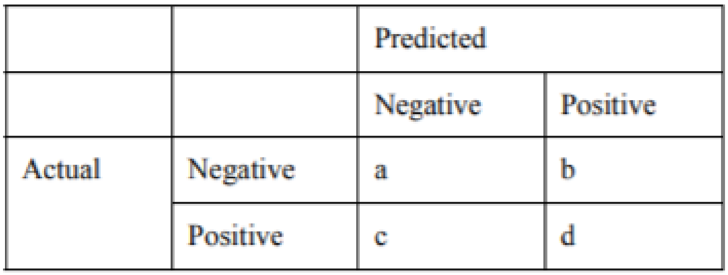
Figure 6: The softmax activation function

2.3 Model Validation & Evaluation

All the models that were created using the algorithms were validated by using Cross Validation. Cross Validation is a validation technique to assess how the results of a statistical analysis model will generalize to an independent dataset. This is done by splitting up the dataset in a training- and validation set. The dataset has been split up in a training/validation set, where 70/30 was used for RNN and LR and 80/20 for SVM and KNN. The algorithms were trained by the training data. After the training process, the algorithms were validated using the remaining (validation) data. This way, the models are validated on data that it has never seen before.

The validation set contains data for each month of every year and data for a whole day. By extracting the whole day and applying it to the predicted model, the model can give an output of the predicted target. Knowing what the targets are from each dwelling in the validation set, it is possible to validate the predictions with the true targets.

In order to evaluate the models, evaluation metrices were used to produce classification reports which indicates which of the models work the best for this specific problem. The confusion matrix (figure 7) contains information about actual and predicted classifications done by the classification system. Performance of such systems is commonly evaluated using the validation data.



		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

Figure 7: Confusion matrix

The outputs in the confusion matrix have the following meaning:

- a is the number of correct predictions where the actual value negative is,
- b is the number of incorrect predictions where the actual value positive is,
- c is the number of incorrect of predictions where the actual value negative is, and
- d is the number of correct predictions where the actual value positive is

The following metrics can be calculated with the outputs of the confusion matrix:

- The accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation: a + d / (a + b + c + d)
- The recall or true positive rate (TP) is the proportion of positive cases that were correctly identified, calculated using the equation: d / (c + d)
- The false positive rate (FP) is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation: b / (a + b)
- The true negative rate (TN) is defined as the proportion of negatives cases that were classified correctly, as calculated using the equation: a / (a + b)
- The false negative rate (FN) is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation: c / (c + d)
- The precision (P) is the proportion of the predicted positive cases that were correct, as calculated using the equation: d / (b + d)

3 Results

The accuracy from results of all the four algorithms are shown in a bar-chart (figure 8).

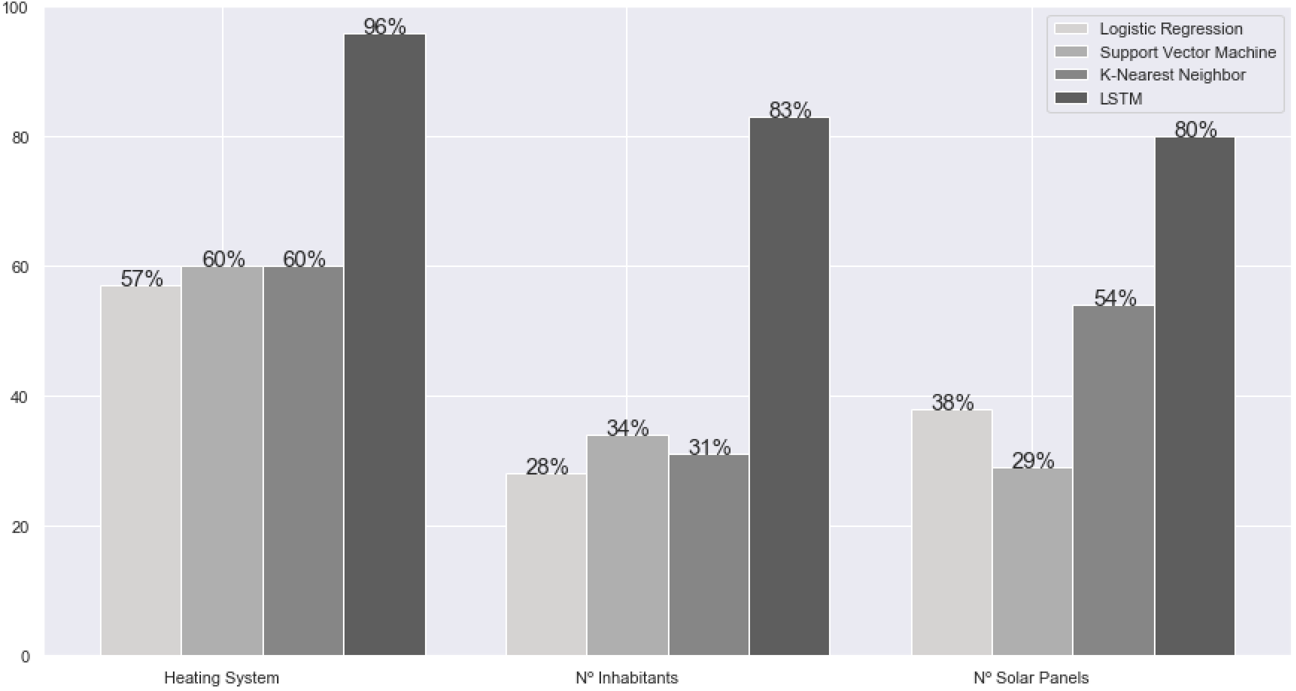


Figure 8: Accuracy bar-chart

As seen in the bar-chart, the first three algorithms have a pretty low accuracy compared to the results of the LSTM.

The machine learning algorithms LR, SVM and KNN cannot give higher accuracy than 60% on the training set to classify the characteristics. This is because LR can only look at one layer of patterns, SVM has limited predictions on big datasets and KNN looks at only one row at a time. The LSTM works with internal memory, therefore makes deep learning possible. The LSTM can classify multiple rows as a pattern, rather than classifying a single row.

When comparing the recall and precision, the results indicate that the LSTM model outperforms the other models. The LSTM model had a recall and precision of 96% when predicting the heating system type, whereas the LR was the lowest performing model with a recall and precision of 28% and 23% when predicting the number of inhabitants. A summary of the classification metrics is shown in table 1.

Table 1: Summary of classification metrics

Algorithm	**Predicted Target**	**Accuracy**	**Recall**	**Precision**
Logistic Regression	Heating system	57%	57%	47%
Logistic Regression	Number of solar panels	38%	38%	33%
Logistic Regression	Number of inhabitants	28%	28%	23%
K-Nearest Neighbours	Heating system	60%	62%	61%
K-Nearest Neighbours	Number of solar panels	54%	54%	54%
K-Nearest Neighbours	Number of inhabitants	31%	32%	29%
SVM	Heating system	60%	60%	59%
SVM	Number of solar panels	29%	29%	26%
SVM	Number of inhabitants	34%	36%	33%
LSTM	Heating system	96%	96%	96%
LSTM	Number of solar panels	83%	80%	80%
LSTM	Number of inhabitants	80%	83%	83%

4 Discussion

The machine learning algorithms Logistic Regression (LR), Support Vector Machine (SVM) and K-Nearest Neighbour (KNN) perform significantly worse than Long Short-Term Memory (LSTM). Since LR, SVM and KNN only look at one row of data, they can only classify the data per row. This means that every row of 15-minute data is classified individually. The assumptions are that the models will not be able to classify the data based on individual 15-minute data rows accurately, because this data provides too little information.

On the other hand, LSTM uses 92 rows of 15-minute data. This provides the model a comprehensive look at the data, which allows the model to classify the data significantly more accurate. LSTM uses long time series data memorization in order to classify the data. With this data, the LSTM classifies 92 rows as a pattern instead of classifying a single row.

During the research, several approaches were made to improve the results on LR, SVM and KNN. This was done by aggregating the data in daily and weekly resolutions. This allows the models to classify the data more accurately, because it is easier to differentiate data on daily or weekly resolutions instead of 15-minute data. For instance, the energy use in a winter day/week will be probably be significantly higher than a summer day/week. For SVM in special, it was mandatory to aggregate the data, because SVM does not support big datasets, so the 15-minute dataset was too large to efficiently work with.

After the first iteration of implementing the RNN, the model classified the data with only one row of data which didn't showed promising results. The input data was then reshaped, so the data contained 92 rows of data which showed only after a couple iterations much better results than before. However, the results weren't satisfying yet. In order to get the get better results, the architecture of the LSTM was changed to a simpler model, by removing one hidden layer and removing a dropout. The model improved significantly, where the model classifies the heating systems with 95% accuracy, the number of inhabitants with 80% accuracy and the number of solar panels to 75% accuracy. Furthermore, the loss function was then changed multiple times for each possible classification. To classify the number of inhabitants better, the loss function "binary_crossentropy" was used and to classify the number of solar panel, the loss function "categorical_crossentropy" was used. This concludes into the results mentioned in the previous chapter.

The LSTM performs best when it comes to classify the type of heating system. The number of inhabitants is difficult to classify, since every inhabitant has their own energy use pattern and hours spend in the dwelling, as well as types of devices used. A single inhabitant can spend as much energy in a week as a family of three. The number of solar panels is also a bit more difficult to classify, because smart meter data only provides the produced energy minus the used energy in the dwelling.

5 Conclusion

This paper compares several machine learning algorithms to classify 33 different dwellings from a neighbourhood called Groene Mient based on the following characteristics: heating system type installed, number of inhabitants and number of solar panels installed. Classifications were done with a daily resolution for Logistic Regression and Support Vector Machine and weekly resolution for K-Nearest Neighbours. For the LSTM, 15-minute resolution was used to feed the model with samples of 92 rows (1 day), by using the energy delivery, consumption, KNMI data and several dummy variables as features. As it is a classification problem, the models have been applied in the following order: Logistic Regression, Support Vector Machine, K-Nearest Neighbours and Long Short-Term Memory.

LSTM performed best compared to the other algorithms for all the target variables (96% predicting heating systems, 83% predicting number of inhabitants and 80% predicting number of solar panels). Due to these results, it is possible to reply to the main research question, since they have been able to predict accurately the different characteristics from the dwellings. These results can be used on policy decisions in order to predict which type of heating installation a dwelling has; the number of inhabitants and the number of solar panels are installed.

Further studies should focus on exploring the possibilities of getting more insight from the dwellings by using datasets with a smaller time interval which allows the LSTM model to perform better. With the purpose of improving the evaluation metrics, it would be possible by using more sample dwellings and resampling the actual data, which magnify the dataset. This is substantiated on the variance between the validation and train loss of each algorithm. Improving the smart meters collecting method (preventing outliers and blank gaps) will help with the recognition of human patterns and dependencies on outside weather conditions. Alongside this, more features of the dwellings can be used to improve the accuracy of the all the algorithms.

Reference

1. Rijksoverheid (2016). *Energieagenda: naar een CO-arme energievoorziening.* Consulted from https://www.rijksoverheid.nl/doe-mee/documenten/kamerstukken/2016/11/21/kamerbrief-over-uitkomsten-energiedialoog

2. EBN (2019). *Energie in Nederland.* Consulted from https://www.energieinnederland.nl/

3. Frank Dixon (2007). *Sustainable Systems Implementation Building a Sustainable Economy and Society.* Consulted from https://globalsystemchange.com/sustainable-systems-implementation-building-a-sustainable-economy-and-society/

4. Erwin Mlecnik (2011). *Defining nearly zero-energy housing in Belgium and the Netherlands.* Consulted from https://ec.europa.eu/energy/en/content/nzeb

5. Pol Van Aubel & Erik Poll (2019). *Smart metering in the Netherlands: What, how, and why.* Consulted from https://www.sciencedirect.com/science/article/pii/S0142061518335142

6. Eoghan McKenna, Ian Richardson, Murray Thomson (2012). *Smart meter data: Balancing consumer privacy concerns with legitimate applications.* Consulted from https://www.sciencedirect.com/science/article/pii/S0301421511009438

7. George Papachristos (2014). *Household electricity consumption and CO2 emissions in the Netherlands: A model-based analysis.* Consulted from https://www.sciencedirect.com/science/article/pii/S0378778814008160

8. Bedir, M. Kara, E.C (2017). *Behavioral Patterns and Profiles of Electricity Consumption in Dutch Dwellings.* Consulted from https://www.sciencedirect.com/science/article/pii/S0378778817306011

9. Chem Cent J. (2012). The smart meter and a smarter consumer: *quantifying the benefits of smart meter implementation in the United States*. Consulted from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3332254/

10. Ning Lu, Pengwei Du, Xinxin Guo, Frank L. Greitzer (2012). *Smart meter data analysis.* Consulted from https://ieeexplore.ieee.org/abstract/document/6281612

11. Gordon Rausser (2017). Smart meters and household electricity consumption: *A case study in Ireland.* Consulted from https://journals.sagepub.com/doi/full/10.1177/0958305X17741385

12. Brian de Keijzer, Pol de Visser, Víctor García Romillo, Víctor Gómez Muñoz, Daan Boesten, Megan Meezen, Tadeo Baldiri Salcedo Rahola (2018). *Forecasting residential gas consumption with machine learning algorithms on weather data.* Consulted from https://www.researchgate.net/publication/334114428_Forecasting_residential_gas_consumption_with_machine_learning_algorithms_on_weather_data

13. KNMI (2019). *Daggegevens van het weer in Nederland.* Consulted from https://www.knmi.nl/nederland-nu/klimatologie/daggegevens

14. Rubén Pérez-Chacón1, José M. Luna-Romera, Alicia Troncoso1, Francisco Martínez-Álvarez and José C. Riquelme (2018). *Big Data Analytics for Discovering Electricity Consumption Patterns in Smart Cities.* Consulted from https://www.researchgate.net/publication/323845683_Big_Data_Analytics_for_Discovering_Electricity_Consumption_Patterns_in_Smart_Cities

15. Thomas G. Dietterich (2002). *Machine Learning for Sequential Data: A review.* Consulted from https://link.springer.com/chapter/10.1007/3-540-70659-3_2

16. Issam El Naqa, Martin J. Murphy (2015). *What is Machine Learning?* Consulted from https://link.springer.com/chapter/10.1007/978-3-319-18305-3_1

17. Applying LSTM to Time Series Predictable Through Time-Window (2002). Consulted from https://link.springer.com/chapter/10.1007/978-1-4471-0219-9_20

18. Improving Electric Energy Consumption Prediction Using CNN and Bi-LSTM (2019). Consulted from https://www.mdpi.com/2076-3417/9/20/4237/htm

19. Understanding Dropout | Part of: Advances in Neural Information Processing Systems 26 (NIPS 2013). Consulted from http://papers.nips.cc/paper/4878-understanding-dropout