UNIVERSITY OF THESSALY

DIPLOMA THESIS

Heterogeneous data for machine learning: The case of load forecasting

Author:
Pavlos LOGDANIDIS

Supervisor:
Manolis VAVALIS

A thesis submitted in fulfillment of the requirements for the degree of Diploma

in the

Department of Electrical and Computer Engineering

October 18, 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Διπλωματική Εργασία

Ετερογενή δεδομένα για μηχανική μάθηση: Η περίπτωση της πρόβλεψης φορτίου

Συγγραφέας:
Παύλος Λογδανίδης

Επιβλέπων:
Μανώλης Βάβαλης

Μια διπλωματική εργασία που υποβλήθηκε για την συμπλήρωση των προηποθέσεων για την απόκτηση Διπλώματος

στο

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

18 Οκτωβρίου 2019

Declaration of Authorship

I, Pavlos LOGDANIDIS, declare that this thesis titled, "Heterogeneous data for machine learning: The case of load forecasting" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Abstract

Pavlos LOGDANIDIS

Heterogeneous data for machine learning: The case of load forecasting

Heterogeneous data are any data with high variability of data types and formats. They are possibly ambiguous and low quality due to missing values and high data redundancy. In order to integrate and make use of all the available information that exists in multiple data sources, traditional machine learning techniques are usually not enough. Data fusion is the process of integrating multiple data sources to produce more consistent, accurate, and useful information than that provided by any individual data source. In this thesis, we first go through some of the necessary definitions associated with heterogeneous data in machine learning as well as data pre-processing methods that are often required when dealing with such data. Next, we go through the most frequently used methods and algorithms with specific examples from other studies, in the depth that we thought was necessary. Specifically, we investigate kernel methods, deep learning, ensembles and other methods that are used less frequently. The real world application that interests us in this thesis is short-term load forecasting. After briefly introducing the problem, we look at how heterogeneous data can be used for enriching the feature sets that are traditionally used in short term load forecasts.

Περίληψη

Παύλος Λογδανίδης

Ετερογενή δεδομένα για μηχανική μάθηση: Η περίπτωση της πρόβλεψης φορτίου

Τα ετερογενή δεδομένα είναι δεδομένα που παρουσιάζουν πολλές διαφορετικές μορφές. Πιθανόν θα είναι ασαφή και χαμηλής ποιότητας λόγω τιμών που θα λείπουν και υψηλής παρουσίας πλεοναζόντων τιμών. Οι παραδοσιακές τεχνικές μηχανικής μάθησης δεν αρκούν για να αξιοποιήσουμε και να χρησιμοποιησουμε όλη την πληροφορία που υπάρχει σε πολλές πήγες δεδομένων. Data fusion ονομάζεται η διαδικασία ενοποίησης πολλών διαφορετικών πηγών δεδομένων για να παραγάγουμε πληροφορία που θα είναι πιο χρήσιμη, με μεγαλύτερη συνεπεία και ακρίβεια από οποιαδήποτε μοναδική πηγή. Σε αυτήν την εργασία, πρώτα θα δούμε κάποιους αναγκαίους ορισμούς που σχετίζονται με τα ετερογενή δεδομένα στην μηχανική μάθηση και τις μεθόδους που χρησιμοποιούνται για να τα προεπεξεργαστούμε. Στην συνεχεία παρουσιάζονται οι συχνότερες μέθοδοι και αλγόριθμοι αντιμετώπισης ετερογενών δεδομένων, από σχετικές έρευνες, στο βάθος που θεωρήσαμε καλύτερο. Ποιο συγκεκριμένα, ερευνήσαμε μεθόδους kernel, βαθιάς μάθησης, ensemble άλλα και άλλες που χρησιμοποιούνται λιγότερο. Η εφαρμογή που επικεντρωθήκαμε είναι η πρόβλεψη ζήτησης φορτίου για μικρο χρονικό διάστημα. Αφού παρουσιάσουμε το πρόβλημα, ερευνούμε τρόπους με τους οποίους θα ήταν εφικτό να εμπλουτιστεί το σύνολο χαρακτηριστικών που παραδοσιακά χρησιμοποιείται για αυτήν την πρόβλεψη.

Acknowledgements

Θα ήθελα να ευχαριστήσω θερμά τους γονείς μου για την αγάπη και την στήριξη τους. Δεν μου είναι εύκολο να εκφράσω με λόγια πόσο τους εκτιμώ και τους αγαπώ. Στη συνέχεια, θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής μου εργασίας, τον κύριο Εμμανουήλ Βάβαλη για τις έμπειρες συμβουλές του και την καθοδήγηση του, καθώς επίσης και τον κύριο Ηλία Χούστη, ο οποίος με την εθελοντική συνεισφορά του μου πρόσφερε απλόχερα πολύτιμες γνώσεις. Τέλος, ευχαριστώ τους φίλους μου για την στήριξη τους όλα αυτά τα χρόνια και ιδιαίτερα τον Στέλιο Γκουντουβά για τις πολύτιμες συμβουλές του για την περάτωση της παρούσας διπλωματικής εργασίας.

Contents

List of Figures

Chapter 1

Introduction

The most commonly used data in machine learning tasks is structured data. This refers to data which is in tabular form in either spreadsheets or relational databases (Ramesh, 2019). An important thing to note however, is that structured data is only a small fraction of the total existing data, with some experts estimating that only 5% of the data generated is homogeneous. The other 95% constitutes heterogeneous data, which is data that has a high variability of formats, data types and other forms of heterogeneity. Very often, heterogeneous data has some negative properties like data redundancy, missing values and more, therefore posing a big challenge to machine learning. In order to integrate and make use of all the available information that exists in multiple data sources, traditional machine learning techniques are usually not enough (He, 2017).

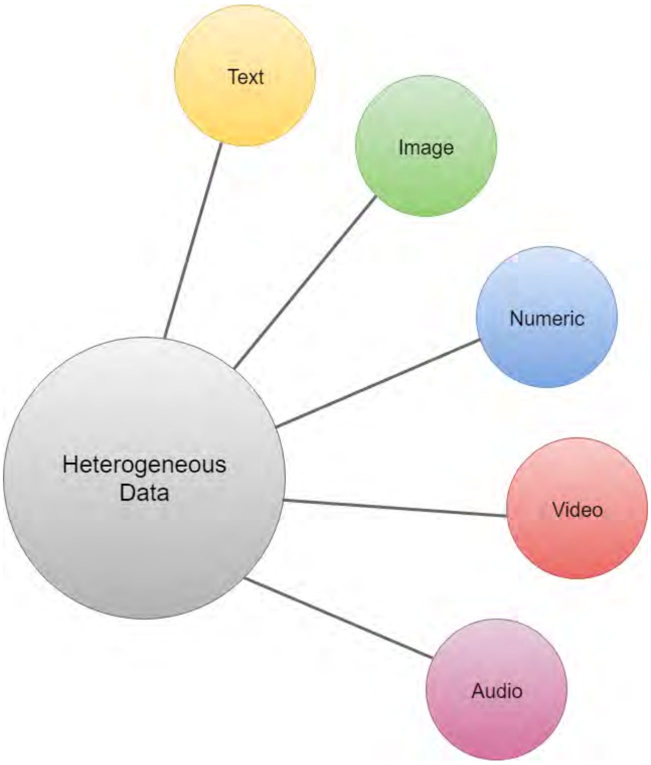

FIGURE 1.1: Multi-modal heterogeneous data.

It is important to develop ways to effectively mine, fuse and learn from heterogeneous data because it contains more information than single-source data since it provides us with different aspects of our target object. It can therefore compensate for missing values and general low data quality of a single data source. “By eliminating the gap between heterogeneous data and fusion of various data sources for correlation analysis, the data could emerge more valuable new information, to achieve the $1 + 1 > 2$ effect" (Zhang et al., 2018). While there are many types of heterogeneity like the difference of spatio-temporal scales and measurement units, we are mainly interested in multi-modality. The term modality refers to data acquisition devices, techniques and frameworks (Lahat, Adali, and Jutten, 2015). Some examples of modalities are image, text, video and audio. Since natural phenomena have very rich characteristics, it is not likely that a single data acquisition framework can provide us with complete knowledge about the phenomenon that we are interested in. By using multiple modalities that are complementary to each other (meaning that each modality brings added value to the whole system), we can expect increases in performance, robustness and other beneficial properties as long as they are handled in a proper way. The multiple forms of multi-modal heterogeneous data are shown in (Fig. 1.1).

The rest of this thesis is organized as follows.

In Chapter 2, we introduce the idea of data fusion from heterogeneous sources and separate fusion methods in two broad categories.

In Chapter 3, we provide some pre-processing steps that are often necessary when working with heterogeneous data.

In Chapters 4 - 7, we review different methods that have been used by researchers to fuse heterogeneous data. Specifically, we look at kernel methods, deep learning methods, ensembles as well as other, less frequently used methods.

In Chapter 8, we introduce the problem of short term load forecasting and investigate the potential benefits of incorporating heterogeneous data for our forecasting models.

In Chapter 9, we provide a synopsis of the previous chapters as well as future research and development prospects.

Chapter 2

Heterogeneous Data Fusion

Data fusion is the process where we combine many heterogeneous sources of data for the purpose of producing more useful, precise and robust information in comparison to only using a single source of data. It so happens that we as humans perform data fusion subconsciously in our day to day lives to perform most of our actions. In order to make sure that our food is edible, we fuse information from our senses like smell, taste and touch. In a similar manner, we rely on fusion of vision and hearing to drive safely and avoid accidents. In these cases as well as many others, our brain performs data fusion using inputs from all our senses and using the combined information it helps us perform complex tasks (Wikipedia, 2019). Technically, multimodal data fusion is the process of combining information from many modalities in order to predict an outcome, whether it's a class through classification or a continuous value with regression (Baltrusaitis, Ahuja, and Morency, 2019). Data fusion can be performed in different stages of the machine learning procedure, as explained below (Haghighat, Abdel-Mottaleb, and Alhalabi, 2016).

2.1 Early fusion

Early fusion is the process of creating new features by integrating multiple features from different data sources (Fig. 2.1). Whether this new feature matrix is going to be effective to our problem or not depends on the quality of the features we choose to merge as well as the fusion method. The most simple feature layer fusion method is to concatenate the features from each data source to form a new feature matrix and use that as input to the preferred algorithm. This method is often unreliable, since the redundancy of information among multi-source data is ignored and may have negative effects in performance (Yann LeCun, Yoshua Bengio, 2015). Early fusion potentially finds correlations between data but often suffers from the curse-of-dimensionality since the combined features are usually large in number.

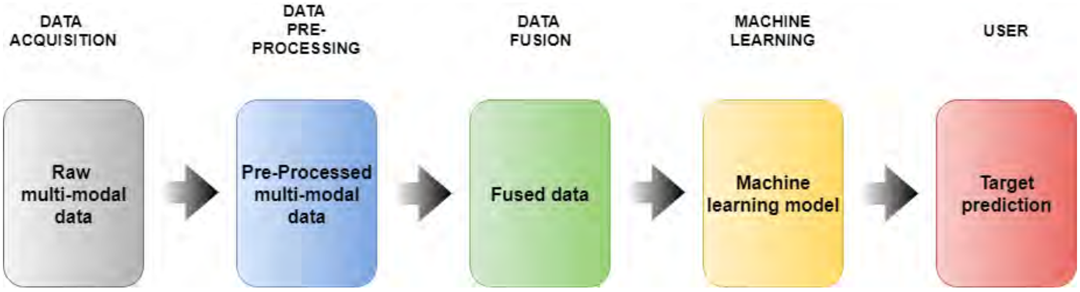

FIGURE 2.1: Early fusion.

2.2 Late fusion

Late or decision layer fusion is when we jointly inference the final result from multiple decisions made by individual models. Each model can handle a single data source so there is no need for feature fusion. Instead, actions like pre-processing, feature extraction and target prediction are made for each data source separately and the final decision is made by taking into account the decisions made by all the models (Fig. 2.2). Therefore, the fusion of information happens at the final level of the ML procedure which is the decision level. Late fusion provides a lot of flexibility since we can use different models for each modality and exploit the advantage of certain models to model certain modalities (e.g. CNN for image data). This method also mitigates the curse-of-dimensionality since the number of features doesn't grow with more data sources but it loses the correlations among the different data sources since they are analyzed independently.

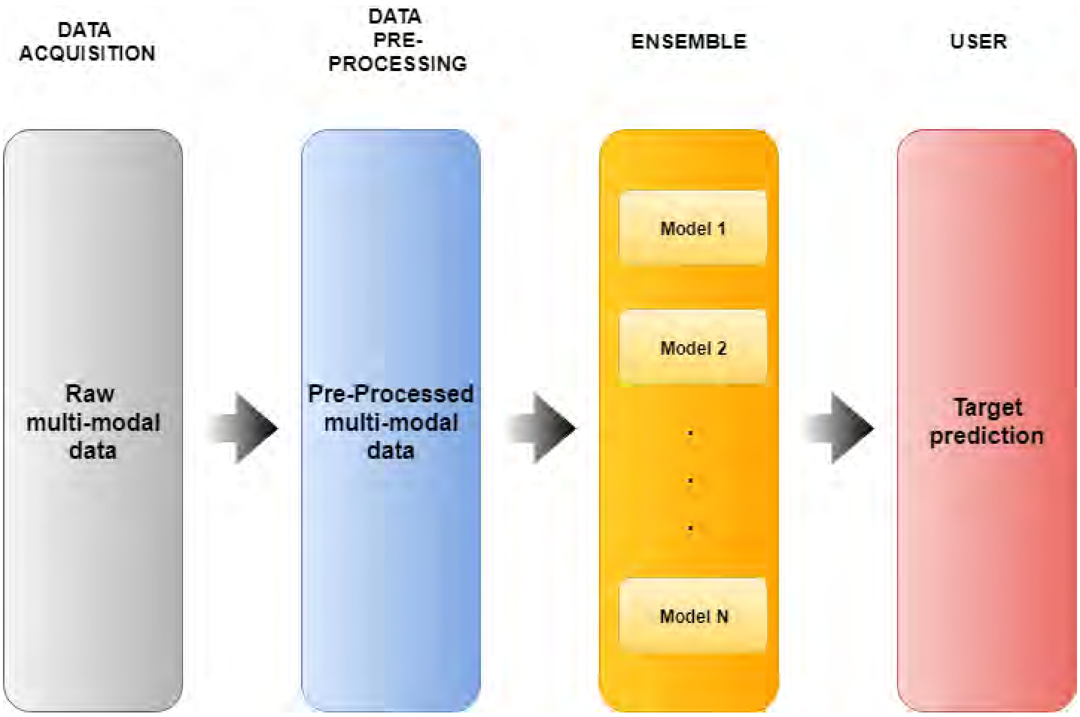

FIGURE 2.2: Late fusion.

We can also combine both approaches, meaning that individual models can be built with early fusion and their decisions are combined with late fusion. This is called hybrid fusion and it potentially exploits the advantages of both early and late fusion that we mentioned above. The most frequent machine learning techniques that we came across in our study are 1) Kernel methods 2) Artificial Neural Networks 3) Ensemble methods and 4) less frequent methods like Bayesian methods, Genetic Programming etc. Some of these techniques can be applied at different stages of the Machine Learning procedure. For example, they can be applied for feature extraction which is the transformation of raw data into features suitable for modeling. They can also be applied for fusion of features and of course their most obvious use, as prediction models.

Chapter 3

Heterogeneous Data Pre-processing

Heterogeneous data usually requires several pre-processing stages in order for it to be useful for machine learning tasks. This is due to some negative effects that are often associated with them such as high data redundancy, missing values and others which impact the quality of the data.

3.1 Data Cleaning

Data cleaning refers to the process of identifying incomplete, unreasonable or inaccurate data and delete or modify such data in order to improve the data quality. We refer to missing values of variables as entries that are empty even though a value actually exists. An easy technique that is often used is simple imputation where we replace the missing value with either the mean, mode or median of the variable. However, when data aren't missing completely at random (MCAR), this method produces biased results. We have MCAR data when there's no relationship between whether a data point is missing and any values in the data set, missing or observed. The missing data are just a random subset of the data. If a large portion of the data is missing, simple imputation might distort correlations between variables, underestimate standard errors or produce misleading p-values during statistical tests. When dealing with most missing value problems, a few good techniques are: 1) Remove the examples with missing variables 2) Fill in missing values by taking into account the similarity between examples for the variables that are available and not missing 3) Fill in missing values by taking into account the correlations between features and 4) Use techniques and tools that are capable of handling missing values.

3.2 Data Integration

Data Integration is a data pre-processing technique that involves combining data from multiple sources, which are stored in various different formats, and providing the users with a unified view of them. Data warehouses are a very common implementation of data integration, which is often called the foundation of effective Business Intelligence and decision making (Arputhamary and Arockiam, 2015). Advanced techniques allow the mixing of structured and unstructured data but they require "clean" data. Technologies like Data Virtualization (DV) and Data lakes are emerging and powerful techniques to deal with data integration on a large scale while at the same time reducing costs and implementation times.

3.3 Dimensionality Reduction

As its name implies, dimensionality reduction is the process of reducing the number of random variables under consideration and can be further divided into subset selection and feature ranking. Subset selection involves finding the optimal subset of the original features by searching over the space of possible subsets. It can be split in 3 different types. 1) Filter approaches where people select the subset of features according to some criteria 2) Embedded approaches where feature selection happens as a separate part of the classification method we are using 3) Wrapper approaches where we use an algorithm e.g. Artificial Neural Network (ANN) to identify the top features. Feature ranking is much simpler since it involves defining a threshold for certain criteria and selecting the features that are above or below it. There are many reasons we might want to perform dimensionality reduction. First, computational challenges often arise with high dimensional data. Second, our algorithm is prone to poor generalization in some cases when the dimensionality is very high. Finally, we can use dimensionality reduction to find meaningful structure in our data, help interpret it and for illustration purposes. A couple of very useful tools for finding underlying reasons for correlations between variables and reducing their number are Factor Analysis and Principal Components Analysis (PCA). PCA can help in several ways when we have many variables and there is some redundancy in them. We can use PCA as a pre-processing tool, a compression tool and in some cases it can be used as a model itself.

3.4 Resampling

It is very common for heterogeneous data to be present in different sampling rates. Even when they are parts of the same system, different measurement devices and sensors often record data at a frequency that is not consistent with the others. This effect is further magnified when we consider the use of external data that is not measured by devices that belong to the system of interest. For example, for the task of load forecasting, we might want to use satellite imagery combined with historical load measurements. It is very likely that their sampling rates are mismatched and this poses a problem for most machine learning algorithms. It is also possible for some data to be irregularly sampled which means it doesn't have a constant frequency. We need our data to be aligned in time before using them as inputs to our algorithms. Depending on the task at hand, we can either upsample or downsample our time series data. With upsampling, we increase the frequency of the data and interpolate the new observations. There are many interpolation techniques like Spline interpolation, Kriging interpolation and others. Downsampling results in data that is in a smaller frequency than the original. We can simply pick the new observations and ignore the adjacent ones or use summary statistics to include information from nearby observations. It all depends on the available data and our specific goals.

Chapter 4

Kernel Methods for Heterogeneous Data

Kernel methods give us a special way to represent different types of data and draw inferences from them. Given a set of arbitrarily complex objects, a kernel method can represent them by using a kernel function $K(a, b)$ which defines how similar any two objects a and b are (Lewis, Jebara, and Noble, 2006). The function takes as input two data objects and returns a similarity value. A small output indicates that the two given objects are not similar and in contrast, a high value indicates that the two objects are similar. The function is also required to be symmetric (since the similarity between a and b is the same as the similarity between b and a) and it should also be positive semi-definite.

This also means that however complex the data object is, we can sufficiently represent a collection of n such data objects, using an nxn kernel matrix. This means that if we are able to define a kernel function for a data type, we can represent a whole dataset of that data type using a kernel matrix. For example, a dataset that consists of 10 numeric vectors where each vector has length 200, can be represented by a $10x10$ kernel matrix which can then be the input of a kernel method like the Support Vector Machine (SVM). The SVM is the most well-known kernel method since it has shown great performance in many complex machine learning tasks and especially classification. Some extensions of the algorithm, allow it to use multiple kernels as a weighted combination, where each kernel is built from a separate dataset. This is very suitable for the study of heterogeneous data in machine learning, since it allows us to use multiple sources and types of data at the same time, as long as we can build appropriate kernel matrices for each data modality.

4.1 Multiple Kernel Learning

Combining kernel matrices is a very intuitive way to fuse heterogeneous data. “When all kernel matrices are formulated as linear kernel matrices, finding coefficients to combine the data sources is directly equivalent to selecting coefficients to combine the kernel matrices built on heterogeneous data sources” (Xu, King, and Lyu, 2007). We start by focusing on each source of data j individually (Society, 2010), and creating the kernel matrix K_j that contains the similarity information between all the samples in that data source. Then, we can design an algorithm that learns the optimal K by “mixing” any number of kernel matrices K_j, for a given learning problem.

This method has plenty of perks. Firstly, we get a homogeneous, standardized input. Secondly, it’s very flexible because of the construction of the individual kernel matrices and the various ways they can be combined. And finally, by finding the optimal mix of kernels, we can ignore information which is irrelevant for a particular learning task. Some examples of Kernel design for non-numerical data are:

- Strings (sequence data) where similarity can be the amount of shared subsequences between two sequences.
- Graphs where the diffusion kernel establishes similarities between vertices of a graph, based on the connectivity information.

For the problem of learning the final kernel matrix, we make use of the property that any symmetric positive semi-definite matrix specifies a kernel matrix and every kernel matrix is symmetric positive semi-definite. Knowing that positive semi-definite matrices form a convex cone which is a good set to optimize over (Lanckriet, 2007), and by defining a cost function to assess the quality of a kernel matrix, we can use Semi-definite Programming (SDP) to learn the optimal parameters. SDP deals with optimizing convex cost functions over the convex cone of positive semi-definite matrices (or a convex subset of it). We therefore gain the advantages of optimizing convex functions, the most important one being that we have a unique optimal solution. This approach is called Multiple Kernel Learning (MKL) and a schematic diagram is shown in (Fig. 4.1).

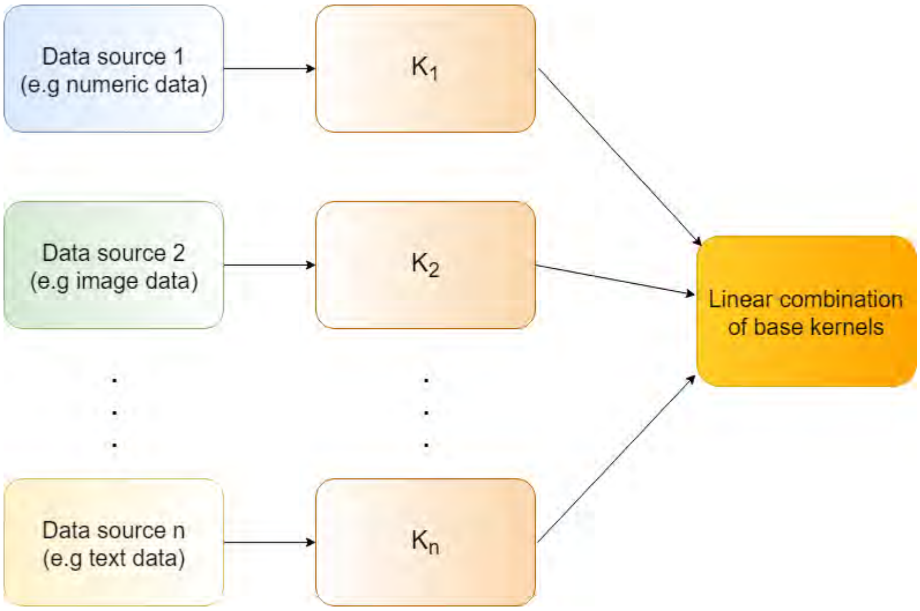

FIGURE 4.1: Multiple Kernel Learning.

Most of the studies we came across, used kernel combination techniques to fuse the different data sources. However, some of them were different and will be mentioned later. Let's take a look at some examples of kernel combination that we found in literature. For the task of web page classification, (Xu, King, and Lyu, 2007) fused multiple data sources from websites that include the text, meta data, title, anchor text etc. using a multiple kernel learning approach. These diverse types of data have different dimensions and properties so simply concatenating them doesn't do much in terms of improving the classification performance. By using kernels however, each data source is represented in a common form which is the kernel matrix. By combining the kernels of each data source as a linear combination, we essentially fuse the information from each source and by using Quadratic Programming, we arrive at a convex optimization problem, which means there is a single best solution that gives us the best classification performance by finding the optimal weights for each kernel. The final kernel matrix can then be used by a kernel method like the SVM. Reference (Lewis, Jebara, and Noble, 2006) investigated whether finding optimal weights for each kernel matrix is efficient, or if we should simply use an unweighted sum of them to form the final kernel matrix. For the weighted kernel case, SDP was used to determine the best coefficients. Their primary conclusion was that a weighted approach is only beneficial when noise is present in the system where the unweighted approach seems to crumble. When there is noise that pollutes some of our data sources, MKL can learn to give smaller weights to those noisy kernels and therefore mitigate their negative impact to the classification performance. Heterogeneous data is used extensively in computational biology. (Lanckriet, 2007) used the MKL approach to predict gene / protein function using the following data: 1) mRNA expression data which is numerical. 2) Sequence data (gene, protein) as a sequence of letters. 3) Protein-protein interaction data in graph format. 4) Others like hydrophobicity data. He derived the appropriate kernel matrices for each data type. In order to do that, he used different similarity measures, depending on the format that the data has. For example, for protein sequence data he used common sub strings between two strings that represent protein sequences. For graph data he used a diffusion kernel, which derives similarity from connectivity information.

In the examples above, a kernel matrix was built from other kernel matrices and was then used by a kernel method like the SVM. We find this to be a very intuitive approach to merge heterogeneous information and it seems applicable in many scenarios, as long as we are able to properly define similarity measures to construct kernel matrices. Next, we will see some other ways that kernels can be used to deal with heterogeneous data.

4.2 Simple Concatenation of features

Some studies use a single kernel so the fusion does not rely on the combination of multiple kernels. Instead, the data is fused before the kernel matrix is derived. The simplest way to achieve that is by concatenating the different features of each data source into a common feature vector and using that to construct the kernel matrix that is going to be used by our classification algorithm. (Long et al., 2018) justify this seamless linking of data from multiple sources with the idea of data interoperability. Their case study involves predicting highway travel time in China for Freeway G5513. Their data consists of weather data, toll data, disposal logs of traffic accidents as well as other time series data. We believe that this approach, while simple can be used in certain cases. However, the type of heterogeneity plays an important role here. While the data they used was heterogeneous in the sense that it came from independent data sources, it was still very easy to encode it in a common form of numeric values. This type of merging would not be applicable in the presence of different data modalities and we would probably be better off by using MKL as we saw above.

4.3 Other methods

In Transfer Learning we make use of secondary data that was not collected to tackle the specific problem at hand, in order to improve our results. We are essentially transferring knowledge that we learned from a task to a related task to improve the learning procedure. It is common for the secondary data to be in different formats than our primary data, so this is naturally a way to deal with heterogeneous data. (Breckels et al., 2016) utilizes a transfer learning framework for classification of proteins. His method makes use of either a k-nearest neighbor or a SVM system to fuse the data sources. For the latter, a separate kernel is used for each of the two data sources and the parameters a are to be determined. The decision function is:

$$f(x, v, a_p, a_A, b) = \sum_{l=1}^{m} y_l [a_l^P K^P(x_l, x) + a_l^A K^A(v_l, v)] + b$$

where K^P is the kernel derived from primary data and K^A is the kernel derived from auxiliary data. This is obviously different from the MKL method that we saw in (Fig. 4.1), where the final kernel matrix is a linear weighted sum of the kernels for each data source. For the experiments of this study, a Gaussian kernel was used.

Reference (Sokolov et al., 2013) utilize a multi-view learning framework to predict the function of proteins. They use both cross-species data and species-specific data as a combination to improve their results. In essence, a separate classifier f is built from each feature set and the label is predicted using inference jointly. If $f^{(c)}$ is the classifier built from the cross-species data and $f^{(s)}$ is the one built with the species-specific data then the final decision is made using the formula:

$$\hat{y} = h(x) = \operatorname*{argmax}_{y \in Y} f^{(c)}(x, y) + f^{(s)}(x, y)$$

In our understanding, this approach can be thought of an ensemble of classifiers even though the authors don't use this terminology. The schematic of the multi-view classifier is shown in figure (Fig. 4.2). In addition to the multi-view method outlined above, they investigate an approach they call the chain classifier. In this approach, the predictions of the cross-species view are used to create features for the species-specific classifier. Their experimental results indicate the superiority of using the two proposed classifiers over the classic approach of using a single piece of data.

In reference (Gönen, 2014) in order to deal with the modelling of heterogeneous data, they map heterogeneous objects from two different domains into a unified embedding space and form an optimization problem where the loss function contains three different scoring functions that give cross-domain interaction and within-domain similarity information between the two domains. Their novel embedding method is called Multiple Kernel Preserving Embedding (MKPE), and its functionality involves projecting heterogeneous data to a latent unified embedding space while preserving within-domain similarities and cross-domain interactions. Gaussian kernels are used for the projection to the new space.

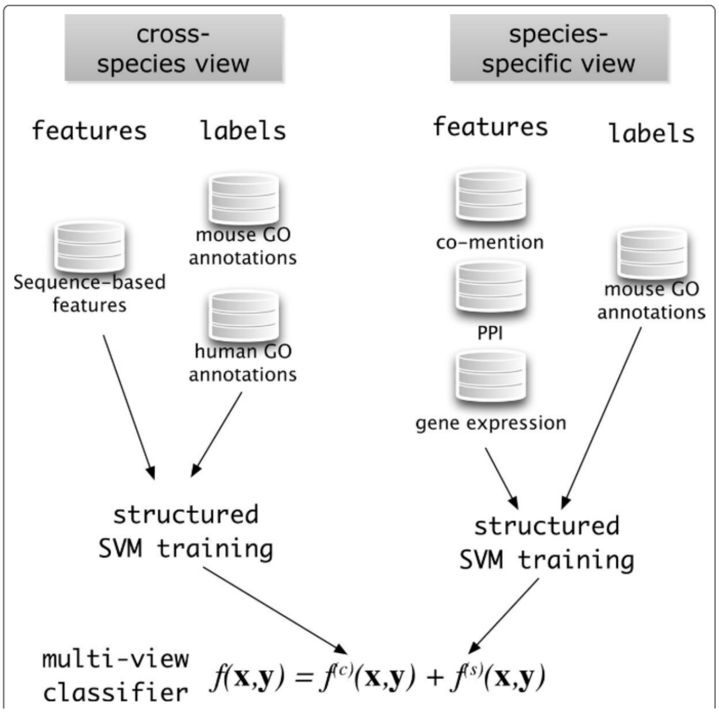

FIGURE 4.2: Multi-view classifier.

Chapter 5

Deep Learning Methods for Heterogeneous Data

Similarly to kernel methods, deep learning has made great advances in dealing with heterogeneous data in machine learning. With Artificial Neural Networks (ANNs), information propagates from layer to layer through multiple non-linear transformations, providing significant results in feature fusion as well as feature extraction. We can use deep architectures in the same spirit that we used Multiple Kernel Learning where every modality is allocated to a unique kernel. Similarly, with ANNs we can assign heterogeneous sub-networks to different modalities, making multi-modal data fusion intuitive and easy. ANNs are very flexible in their architecture and there are many different types like the Deep Belief Network (DBN), the Convolutional Neural Network (CNN) and many more, each of which offers unique advantages in feature fusion and learning.

5.1 Sub-networks for each modality

Due to the layered architecture of deep ANNs, each layer is supposed to provide a more abstract representation of the data, hence we often use the last layer as a way to represent the original data. As we stated above, the most common way of dealing with multi-modal data with deep learning, is to pass each data modality through its own deep sub-network, which is specifically designed for that modality. We then add a hidden layer which is used to project the individual modalities to a common embedding space. This joint multi-modal representation can then be passed though several other hidden layers or we can use it directly for prediction (Baltrusaitis, Ahuja, and Morency, 2019). A disadvantage of using deep learning that is worth mentioning, is the lack of interpretability. Using multiple modalities, it is difficult to know whether one is important or not for the final prediction. Furthermore, we know that deep learning requires huge amount of training data to be successful.

(Zhao, Hu, and Wang, 2015) proposed a framework for feature selection using deep sub-networks combined with sparse group lasso analysis (Fig. 5.1).

They aim to transform the multi-modal data into a unified representation with the use of modality specific sub-networks. Each sub-network is based on Stacked Denoising Autoencoders (SDA) (Fig. 5.2) and have the same objective function, despite of their different structure.

This optimization problem is solved using sparse group lasso, which performs variable selection in a group of variables, and the optimal weights for each feature group are identified. Finally, after ignoring those feature groups with insignificant weights, all the outputs of the sub-networks are joined into a feature vector and

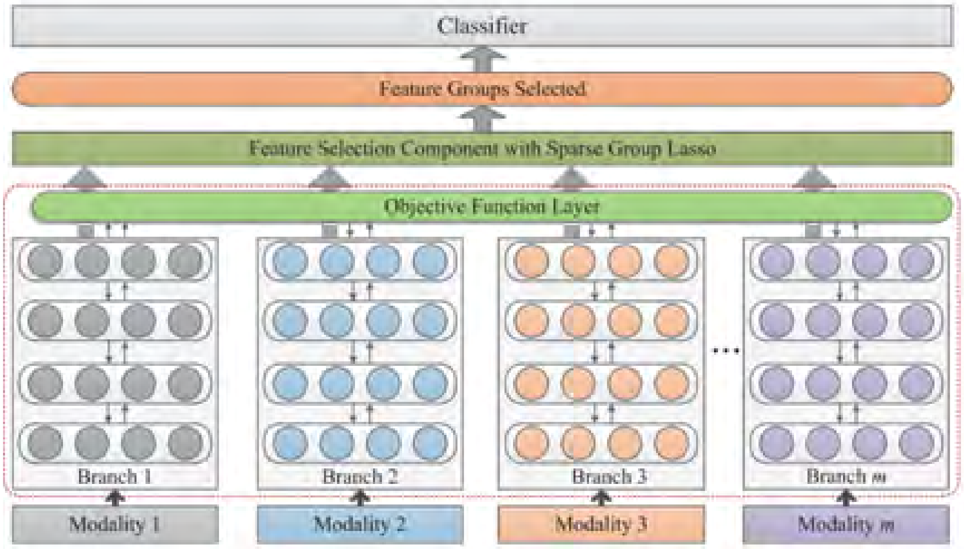

FIGURE 5.1: Architecture overview (Zhao, Hu, and Wang, 2015).

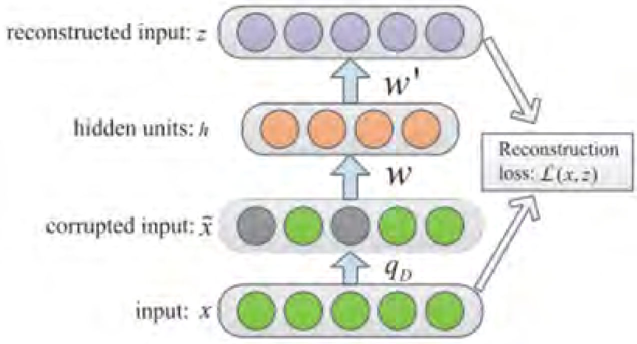
reconstructed input: z
W'
hidden units: h
W
corrupted input: $\tilde{x}$
q_D
input: x
Reconstruction loss: $\mathcal{L}(x,z)$

FIGURE 5.2: Illustration of a Denoising Autoencoder (Zhao, Hu, and Wang, 2015).

a part of their architecture called Feature Selection Component uses this vector to output the final weight vector.

In another study (Srivastava and Salakhutdinov, 2012), the authors try to tackle the problem in a similar way, in other words to learn a representation of the data. They employ a Deep Belief Network (DBN) architecture to learn a joint representation of multi-modal data. The main idea of this architecture is to use separate models for each modality to learn low-level representations and then concatenate them to create the multi-modal input. Furthermore they way they use the DBN allows them to easily handle missing modality data. In the case we want to create a bi-modal image-text DBN, we will encode text as sparse word count vectors and images as real-values dense pixel intensities or extracted features. We will have the image-specific DBN to model the distribution over those extracted features using Gaussian Restricted Boltzmann Machines (RBM) and the text-specific DBN will model the distribution of the word count vectors using Replicated Softmaxes. We form a multi-modal DBN by learning a joint RBM on top of them (Fig. 5.3). The reason to use modality specific models is because each modality has different statistical properties and it is difficult and restrictive to find correlations across them. The use

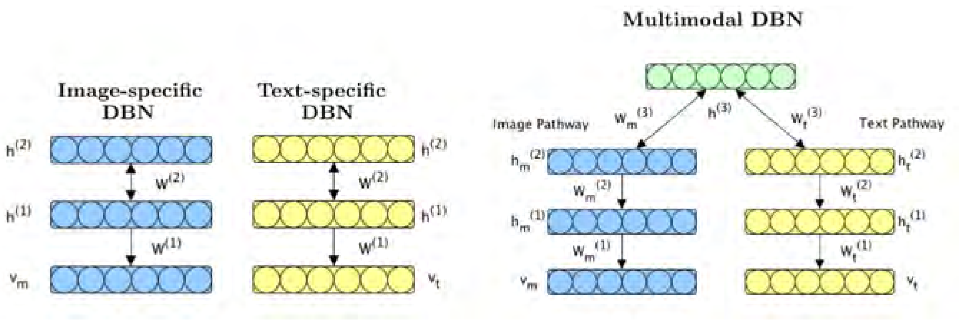

FIGURE 5.3: **Left:** Image-specific and Text-specific two-layer DBN. **Right:** A Multi-modal DBN that models the joint distribution over image and text inputs. (Srivastava and Salakhutdinov, 2012)

of modality specific models allow the data to be represented in a higher level, removing any modality specific correlations which allows to RBM to identify relations easier.

Similarly to the previous studies, (Miech, Laptev, and Sivic, 2018) presents a method for video retrieval from text using a joint text-video embedding. They propose a Mixture-of-Embedding-Experts (MEE) model that can calculate the similarities between different video modalities and text by generating expert weights for the contribution of each modality (Fig. 5.4). This also enables the model to handle missing video modalities. Furthermore the model can also extract information from

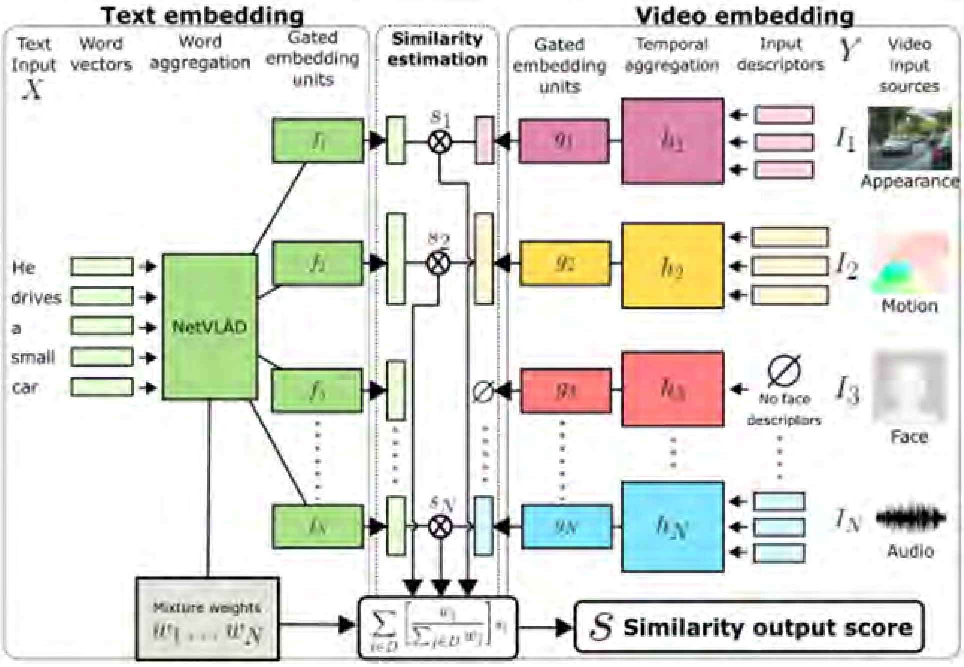

FIGURE 5.4: The Mixture of embedding experts (MEE) model (Miech, Laptev, and Sivic, 2018).

images because it can treat images as motionless and soundless videos (Fig. 5.5). The MME model is easily extensible and can be made to handle other data sources that convey important information for a specific task such as faces in videos from which we can extract emotions, age or gender. Their goal is to learn a common embedding

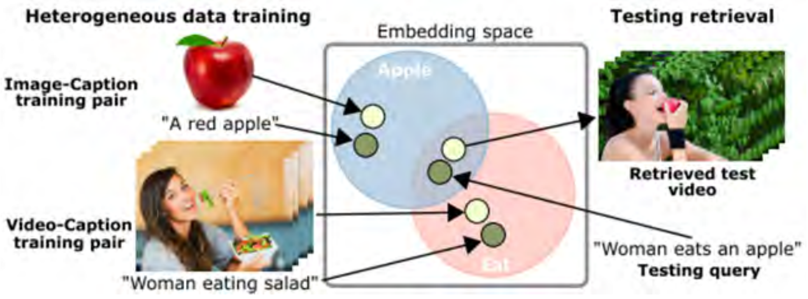

FIGURE 5.5: Obtaining information from both Image-Caption and Video-Caption pairs (Miech, Laptev, and Sivic, 2018).

space for text and video in which a textual and a visual sample will be close only if they are semantically similar. The model requires for each input video some streams of descriptors to exist that represent motion, appearance, audio, or facial characteristics of people but not all of them in all data sources are required. Using a NetVLAD aggregation module (Arandjelovic et al., 2018), we aggregate the text input which is a sequence of word embeddings. Similarly each input stream of the video descriptor is aggregated into a single vector and finally we combine the similarity scores of each text-descriptor pair.

The above studies try to transform the heterogeneous input to a common representation and extract valuable information from this representation, but it is not the only way to fuse heterogeneous data with deep learning. (Audebert, Le Saux, and Lefèvrey, 2017) presents a innovative module for semantic labeling from heterogeneous data using fully convolutional networks (FCN). An unsophisticated approach to fussing heterogeneous data with deep networks is to concatenate all inputs sources and use the outcome as the new input. However this has shown to reduce the performance of the network, reducing the accuracy due to the fact that each source requires different processing. They propose a fusion network that is based on residual correction and use this to combine predictions coming from modality-specific sub-networks that use two types of data (Fig. 5.6). Furthermore they improve this model by introducing a specialized fusion neural network and managed to achieve state-of-the-art results (Fig. 5.7).

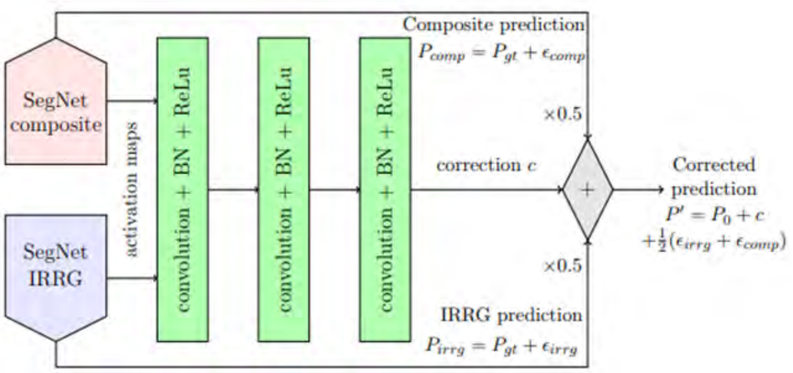

FIGURE 5.6: Fusion network to correct predictions with information from complementary sub-networks (Audebert, Le Saux, and Lefèvrey, 2017) .

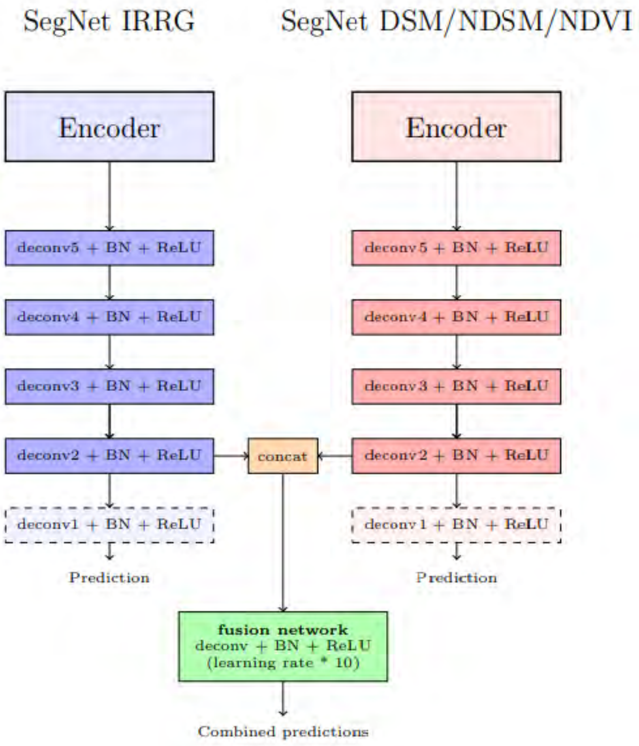

FIGURE 5.7: Fusion network strategy (Audebert, Le Saux, and Lefèvrey, 2017) .

5.2 Concatenation and feature extraction

We can also use heterogeneous data sources to extract features and use these intermediate features for our models. (Zhou et al., 2017) presented a innovative method for feature extraction named Deep Learning Feature Selection (DLFS) that aims to deal with the problems of using heterogeneous data and the constrains of supervised learning. Their goal is to predict if a patient will stay in a hospital more that seven days using only data from the first 24 hours of his admittance to the hospital (Fig. 5.8). One of the biggest challenge they faced is the type of heterogeneity that their data have. There isn't a standard interval where data from the patients were collected and this means that many patients have their data sampled irregularly and the frequency of sampling is not the same. The solution they propose is to interpolate the heterogeneous records by using either Spline Interpolation or Gaussian Process Regression. The next step is to present the data to the DLSF to get a compact

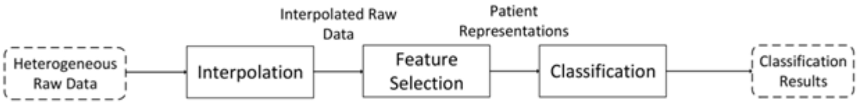

FIGURE 5.8: Overview of the predictive diagnosis framework (Zhou et al., 2017).

representation of each patient. The architecture of the DLSF is implemented using a deep neural network of a stack of denoising autoencoders (SDA) where each layer is a denoising autoencoder (DA) (Fig. 5.9). Finally, at the last step we use a classification algorithm such as Support Vector Machines or Artificial Neural Networks to get the final prediction.

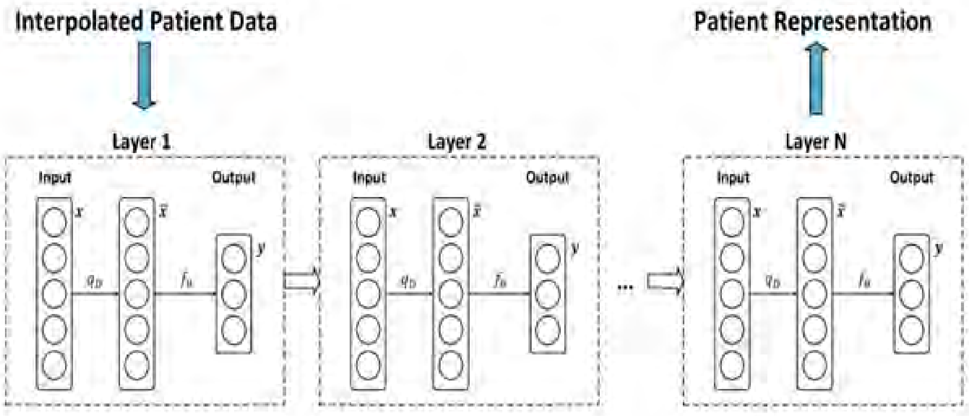

FIGURE 5.9: DLFS feature learning using stacked denoising autoencoders (Zhou et al., 2017).

Along the same lines (Ma et al., 2016), for crowd-sensing, gathered data from various sensors on a smartphone and in combinations with temporal information created an integral feature named context fingerprint. The proposed architecture is a four-layer Deep Belief Network (DBN) with the first layer being visible and the other three hidden (Fig. 5.10). The first layer acts as input and all the data collected at each time point is concatenated in this layer. The other three layers are Restricted Boltzmann Machines responsible to learn the representations of these data. With this model they managed to identify and categorize people on whether they where walking/cycling or using a car/bus.

5.3 Other

The last deep learning studies we are going to look at, are quite different from the ones we have seen so far regarding the way they deal with the heterogeneity of the data.

Reference (Chen et al., 2016), attempted to model traffic accident risk by using big and heterogeneous data (GPS records and traffic accident data). They first transform the traffic accident data as well as the GPS mobility data to a common grid format which is actually enough to make inferences for risk level at a certain region. However, this method doesn't consider areas that are nearby the region of interest, making the model too simple. The inclusion of nearby areas increases the complexity of the problem which is the reason that they use Stacked Denoising Autoencoders to learn latent human mobility features of much reduced dimensionality. Even though it is an interesting study, they deal with the heterogeneity of the data at the pre-processing stage which is something we are not currently interested in.

Finally, (Guo et al., 2019) propose a framework for handling both real-time and heterogeneous data. The main idea is to treat data coming from different sources as a separate batch and use it to train a separate model each time. This model is

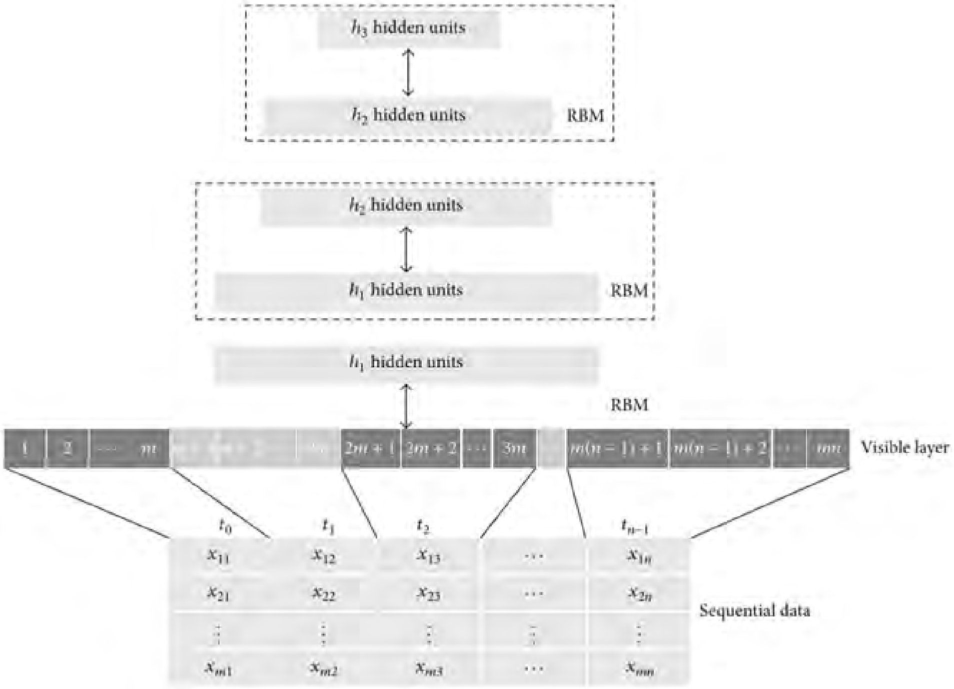

FIGURE 5.10: The architecture of the deep belief network for mobile crowd-sensing (Ma et al., 2016).

then saved in a pool of models, each of which was trained on its own batch of data. This avoids dealing with fusing heterogeneous data at the feature level. The same process takes place for real-time data. At the testing phase, all models are called for discrimination and the final decision is made using a statistical criterion. Testing was made on image data of two different medical types, so the heterogeneity in this case doesn't refer to multimodality, which is what we are mainly interested in. We are skeptical as to how this framework would work on multi-modal data, since the number of inputs changes for models of different modality. Therefore, calling models of different input size for discrimination is impossible, unless all data types are first brought into the same dimensions through pre-processing. In conclusion, we believe that this is an interesting idea but it doesn't seem to work for the type of heterogeneity we are interested in.

Chapter 6

Ensemble methods for Heterogeneous Data

As we mentioned in previous sections, kernel methods and deep learning have been very effective in the fusion of heterogeneous information. In those cases, the integration of multi-source data happens at an early stage of the machine learning process. There is however another route that we did not explore, which is late integration. In this approach, individual data sources are used to derive separate classifiers and those classifiers are then combined to build an optimal final model. This method overcomes difficulties that are present when fusing data at the feature level and when done correctly, it can generate accurate and reliable classifiers. While we only found a few studies on heterogeneous data with late integration, it still is a promising route when working with heterogeneous data in machine learning problems.

The process begins with the construction of the individual models. Each data source/modality is used to construct a separate classifier which provides a lot of flexibility. Some models may work better with certain data types (e.g. CNN for images) and we are able to exploit this property when using ensembles. The decisions made by the individual models are taken into account for the final decision. The intuition is that if each classifier makes different errors, then their strategic combination can reduce the total error. There are many ways to combine predictions and some experimentation may be needed to find the optimal one for a given task. Some popular methods are Bootstrap aggregating or Bagging, Boosting, Bayesian parameter averaging, Stacking and more. The described work flow is shown in figure 6.1.

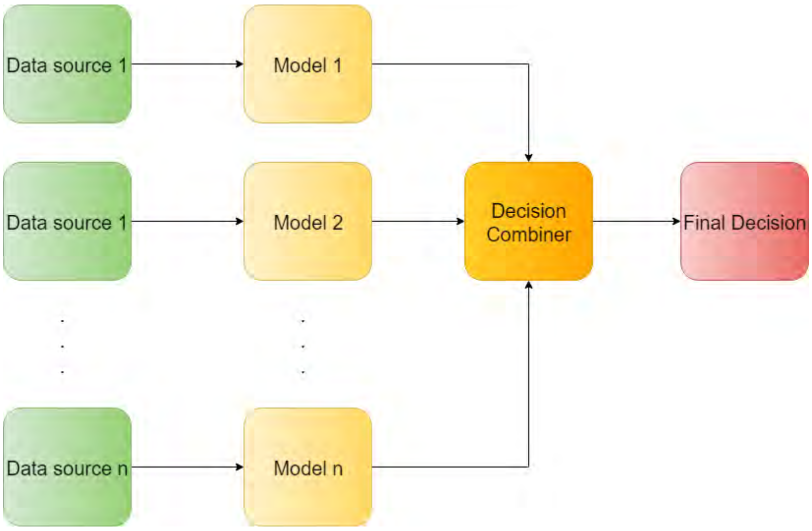

FIGURE 6.1: Ensemble architecture.

Text and image data have been successfully used together using an ensemble scheme in (Peng et al., 2016) for word sense disambiguation and information retrieval. Similarly, for human activity recognition (Chung et al., 2019) uses LSTM neural networks as base learners for different body sensors and uses their predictions as meta-features. These meta-features are taken as input by a meta-learner which is essentially a module that performs different ensemble techniques to arrive at the final decision. Another important note is that we are not limited to a single data source for each classifier. Features from different data sources can also be combined for training the different classifiers of the ensemble as done in (Chatterjee et al., 2015). This provides even more flexibility to the whole process of fusing heterogeneous data.

Ensembles are particularly useful when some of the data sources have missing information which is likely to occur when working with heterogeneous data. Instead of dealing with the missing data at the pre-processing stage as covered in 3, it can also be handled at the decision level. For example, (Aziz and Reddy, 2010) developed a method to improve the decisions made by individual models, using heterogeneous data sources which often contain incomplete information. The algorithm they developed is a modified version of the AdaBoost algorithm called HeteroBoost. The algorithm works with multiple data sources and prioritizes data objects with partial information instead of the common ones. The main idea is that it will give a much higher weight to a data object the less it appears in the data sources.

Ensemble selection is an important step in late fusion and involves selecting the best set of classifiers out of the total generated ones. It can be done statically with Static Ensemble Selection (SES) or dynamically with Dynamic Ensemble Selection (DES). In the fist case, we aim to select the set of classifiers that gives us the best performance on all of the training instances and use these classifiers in a combination at the testing phase. DES one the other hand, focuses on each training instance separately and selects the optimal subset of classifiers to obtain a better result. (Ballard and Wang, 2016) propose a framework of DES with optimization (DESO). The basic ideal of DESO is that different sets of classifiers will have better results in some local regions of our feature space, created from heterogeneous data sources. After generating N classifiers, during training the algorithm will use k-means to cluster the validation set and after running Simulated Annealing for each cluster we chose the classifier with the highest diversity measures. When classifying an instance it is added to the closest cluster and the corresponding sub-ensemble performs the classification. This study uses two non-pairwise diversity definitions: Coincident Failure Diversity and Minority-Failure Diversity.

Chapter 7

Less frequent methods

7.1 Clustering

(Nikolopoulos, 2012) presents a software made to characterize a cluster based on three quality vector elements to characterize customers for an energy company based on their behaviour. One of the biggest problem in smart grid system and smart energy systems is the fact that we have to deal with the problem of finding correlation and extracting information from many variables and most of our data are real time data. Most of our data are heterogeneous and constantly changing, collected from many different sources. Their software follows some steps as setting specific key performance indicators, performs clustering on the data and presents the data to the user. After the clustering results, all clusters are analysed and categorized based on a three element vector containing: 1) Centroid placement, which is the coordinates (x, y) of the final centroid of a cluster. 2) Entropy calculation, which they define as the average statistical dispersion of all Euclidean distances of the cluster, around a centroid. 3) Population variance, which represents the ratio of the measured entropy e of a cluster no. N and the number of n customers that are members of the specific cluster N. Furthermore, they define the similarity between different clusters and cluster sets and by doing this the user is able to make important assumptions about some customers and identify similar trends and patterns among them.

A completely different take on how to unify multiple clusters is presented by (Filkov and Skiena, 2010). They proposed a method to combine microarray datasets, into a unified representation, based on consensus clustering. Consensus clustering is the process where the same clustering algorithm is applied to many datasets and we want to find a single cluster for all datasets. This process is based on the distances of set-partitions which are collections of disjoint sets that cover a set completely. In other words, we can say that consensus clustering has the role of a median between some given clusters and by having this final cluster we can extract information for the dataset used for creating this cluster.

7.2 Bayesian Joint Analysis & Gaussian Processes

In many data science problems when we are working with many disparate datasets, we assume that these datasets are essentially the same and can be represented in a lower-dimension space which is more convenient. However (Ray et al., 2014) claims that this is quite restrictive and a better, more flexible way to solve such problems is to factorize the feature-space into shared components among all the datasets. This paper proposes a Bayesian factor analysis to handle heterogeneous datasets by dividing the feature space into two categories of components, one category is data specific components and the other is the shared components. The model is extended to handle many heterogeneous data sources with spatio-temporal dependencies by employing a kernel stick breaking process (KSBP) that can find shared statistical feature among many data types, managing to take advantage of the data better.

Another problem with heterogeneous data sources is the fact that many data will be irregularly sampled and many models will not work with them unless we make some assumptions about our data. (Ghassemi et al., 2015) proposed a technique to handle irregularly sample clinical data, by using multi-task GP (MTGP) Models to transform them into a new latent space. This model in an extension to multi-task GP models that makes use of the covariance to reduce the uncertainty of the data. The model is tested on a dataset consisting of patient data in combination with clinical notes. The clinical notes are then processed using topic inferenced and are transformed into a 50-sized vector. Finally the only keep 9 topics, the ones having posterior likelihood above or below 5% of the population baseline likelihood across topics. Before applying different machine learning algorithms three matrices are created 1) The admitting SAPS-I score for every patient 2) The average topic membership for the nine identified topics 3) The inferred MTGP hyperparameters across the nine topic vectors The machine learning algorithms are applied to an aggregated matrix created from the three above.

7.3 Other methods

In addition to the above techniques, (Wang et al., 2019) proposes additive partially linear models for handling massive heterogeneous data. These models provide a way to do analysis of massive heterogeneous data using the divide-and-conquer strategy. The proposed framework for modelling massive heterogeneous data can be applied to extract several common features across all sub-populations while exploring heterogeneity of each sub-population. In this paper, the partially linear model (PLM) is generalized and an additive partially linear model (APLM) for modeling massive heterogeneous data is proposed. In essence, a huge dataset is split in sub-datasets where machine learning techniques are applied to them and the results are combined to get the finals result about the whole dataset.

(Korkmaz et al., 2015) created a model to predict civil unrest using different types of indicators in Latin America over the course of 2 years, with data collected from social media, news sources, political databases, Tor statistics and exchange rates. The model used is a Lasso model due to the fact that it also gives us information about the nature of such an event while a simple model with similar performance can’t give us this information. The goal is to predict the day that a nationwide large protest will occur in multiple locations using the following features 1) daily counts of events, 2) average intensity of the events in ICEWS, 3) average tone of the daily events in GDELT (aimed to measure the general sentiment of the entities involved in the event) 4) Goldstein scale score of daily events in GDELT (a collaboration score assigned to each event; the higher the score between the two actors, the greater their collaboration).

Chapter 8

Heterogeneous data for short term load forecasting

Load forecasting is a very important task for the electric power industry (Gupta, 2017). Its purpose is to estimate future load demand by using various data like historical consumption data and other external data from different factors that contribute to the consumption of electrical energy. Electric companies require accurate future load predictions in the short term (few minutes to 7 days) as well as the medium (1 week to 1 year) and the long term (more than 1 year). They can leverage this knowledge to appropriately purchase and generate electricity and for other important decisions like load switching, infrastructure development and contract evaluation. Different types of load forecasting are needed for different types of decisions of a power utility. For example, short term load forecasting is needed for the effective and profitable management of everyday operations in electrical utilities and unit commitment. Many operating decisions rely on short term forecasts. Accurate forecasts lead to a more reliable system as well as economical savings. Medium term load forecasting is used for fuel supply scheduling and unit maintenance. Finally, long term load forecasting is used to predict the need for expansion, staff hiring and purchases of new equipment.

Different determining factors are usually considered for short-term load forecasting than medium and long term load forecasting. Our focus is on Short-Term Load Forecasting (STLF) which is mainly affected by the following variables 1) Time is very important because consumer load changes a lot depending on the time of day, day of week, week of month and month of season. 2) Economic factors like the price of electricity also play a vital role on the average load of the power system. 3) Weather affects electricity use dramatically. It includes measurements like temperature, humidity, cloud cover, wind speed and others. 4) Behavioral patterns of consumers which are highly variable throughout the day and are affected by a multitude of different factors including occasional social events like concerts.

Some very influential factors of medium and long term load term load forecasting are 1) Customer appliances' description 2) House sizes 3) Equipment age 4) Technology changes 5) Employment levels 6) Customer behavior 7) Population dynamics 8) Economic factors 9) Electricity prices

Short term load forecasting is performed using different methods, depending on the data that is being used. Statistical methods focus only on the time series of past load demand measurements and extrapolate predictions in the future. Expert systems use rules and procedures that are defined by human experts in load forecasting. These rules are used by software that can make forecasts without using assistance from humans. Another class are the regression/causal methods. Instead of extrapolating the time series of past load values, we make the assumption that there is a cause-effect relationship between various parameters (such as weather parameters) and the load. We then try to develop a representation of these causal relationships using machine learning models. We can also combine both approaches, meaning that we model the time series characteristics of load in combination with causal factors. In order to make use of heterogeneous data, we obviously have to focus on the last two classes of the methods we mentioned.

8.1 Peer review of STLF with heterogeneous data

Since load demand is affected by multiple factors, it is important to consider as many of them as possible to achieve better forecasting accuracy. Data from these factors very often reside in different databases and in different formats. Since we want to use this data as input to our machine learning algorithms, we must encode all different data types into a common representation that can be handled by our models. Most of the studies we came across, refer to calendar and weather data with the combination of historical data as heterogeneous data. While this data does in fact have characteristics of heterogeneity, we find it very interesting when different data sources have different modalities like text, video, numeric etc. Data fusion of these different data representations can provide us with more contextual information about the current state of the demand side and improve predicting performance.

8.1.1 Weather, calendar and historical data

In (Cheng et al., 2019), they use 3 categories of data i.e. historical load demand data, calendar information and weather information to predict future load demand. There are n + 18 total features where n are from the historical data, 13 weather features and 5 calendar features. To get a small number of n, not all time points are considered but only the ones that are highly correlated with the target which is load demand. Auto Correlation Function is used for this purpose. To encode the historical load data, an LSTM recurrent neural network is used and this encoding is later merged with another branch that fuses weather and calendar information with a feed forward neural network. They call their architecture Powernet. The schematic is shown in (Fig. 8.1).

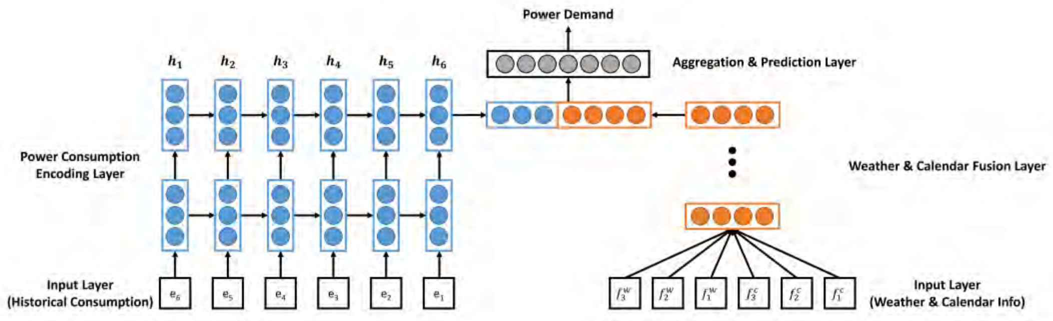
Power Demand
h_1
h_2
h_3
h_4
h_5
h_6
Aggregation & Prediction Layer
Power Consumption Encoding Layer
Weather & Calendar Fusion Layer
Input Layer (Historical Consumption)
e_6
e_5
e_4
e_3
e_2
e_1
f_3^W
f_2^W
f_1^W
f_3^C
f_2^C
f_1^C
Input Layer (Weather & Calendar Info)

FIGURE 8.1: Powernet architecture.

Similarly (Tong et al., 2018) aims to use heterogeneous data which consists of historical load measurements, weather parameters and date parameters in order to train deep NNs for load forecasting purposes. The base architecture of the neural network is the SDA or Stacked Denoising Autoencoder. The system which is depicted in (Fig. 8.2), consists of two modules: a feature extraction module and a load forecasting module. For the feature extraction module, they use 3 different branches of neural networks. The first branch takes as input the previous days (k-1) load measurements every hour so a total of 24 features. The next one is a concatenation of load measurements for days k-7, k-14, k-21 which are the measurements of load of the same day for the past 4 weeks. That would be 72 more features. The last branch contains the weather factors. Along with the extracted latent features of the 3 branches, the season parameter is also concatenated to form the final feature vector which will be passed to the load forecasting module. Forecasting is implemented using support vector regression (SVR).

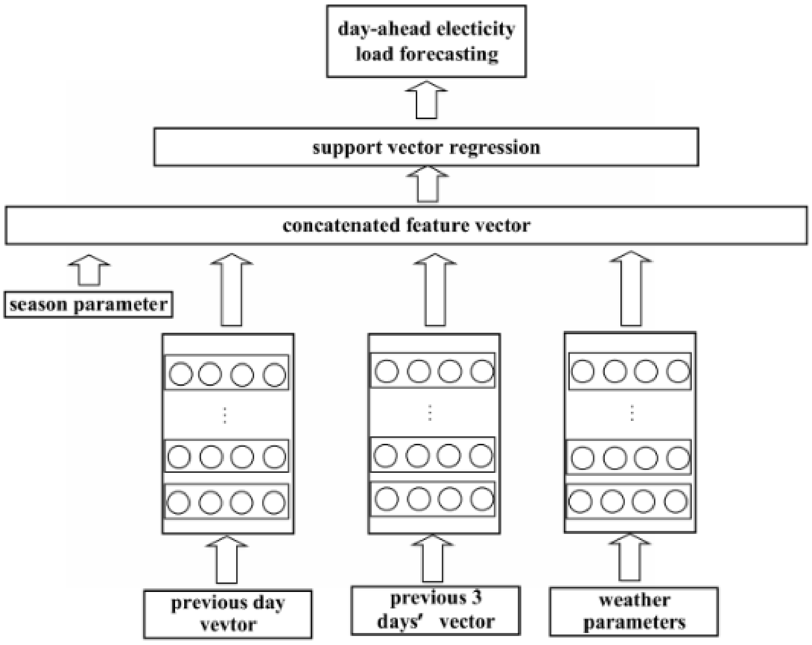
day-ahead electicity load forecasting
support vector regression
concatenated feature vector
season parameter
previous day vevtor
previous 3 days′ vector
weather parameters

FIGURE 8.2: Neural network architecture in (Tong et al., 2018).

In the last study with this combination of input data (Wang et al., 2018), they first separate customers into 3 categories using k-means clustering because their load time series consumption is quite distinct (residents, industry and institutions). The load forecasting is done for individual customers, which is a difficult problem because of high volatility issues. The neural network architecture consists of two parts, one for the temporal feature extraction of the load historical data and one for the external factors like weather, temperature and date. GRU recurrent neural networks are used for the temporal load features. There is a merge layer that fuses the multi-source extracted features. External features are also numerically encoded and normalized. The schematic diagram of the network is shown in (Fig. 8.3). To help with the vanishing gradient problem, a batch normalization layer is added after the merging layer, which is followed by two dense layers and the output layer.

8.1.2 Incorporation of social media data

Social media data like Twitter posts can be used to extract useful information about peoples' activities and behaviors. Load demand is highly affected by these factors so if we mine this knowledge from social media data, we can combine it with other

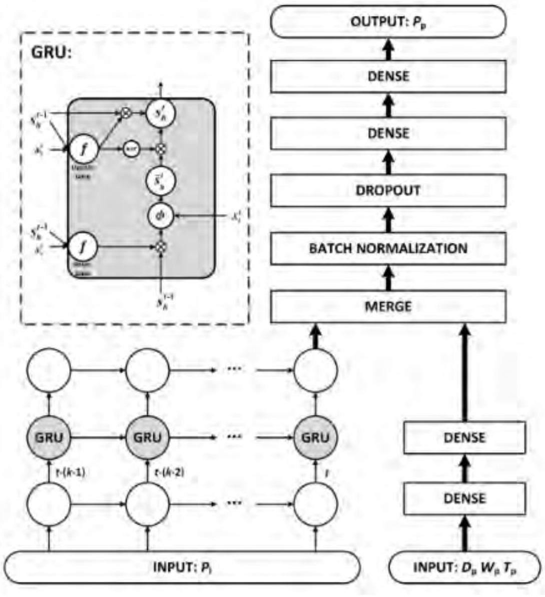

FIGURE 8.3: GRU neural network architecture in (Wang et al., 2018).

data sources to perhaps predict future power needs more accurately. For example, announcements of conventions and concerts that can be found in blogs, local websites and social media can affect electrical power usage in the grid.

Human activity can be captured with social media data in different ways. Some studies use only the volume of posts for a given time period to estimate electricity demand. (Luna et al., 2017) use tweet counts every half hour to predict future load in Peru for the short-term. They verify that there is a correlation between the volume of posts and electricity needs. In a similar study, (Deng et al., 2018) show the potential of using social media data from Twitter as a way to represent human activities in order to model consumption of electricity at building-level.

Traditional ML methods mainly use historical data to forecast future load demand because the time series of load consumption shows characteristics like trend and seasonality which can be adequately modeled using historical data for large-scale power forecasting. However, these methods don't work very well at the building level because there are less temporal patterns that can be detected in the data of a smaller area. To address this, this study uses geo-tagged tweets as the data source to forecast future electricity consumption. The study was done in buildings of a campus in New York. The idea is that the number of tweets indicates that there might be more people and/or more activity like heating, lighting, air conditioning and other human activities that require electrical power.

These two studies used the volume of tweets as their data source and ignored what the posts actually contained. (Stavast, 2014) in his thesis, attempts to mine social media data and use them for the prediction of energy. Techniques that use social media have been successful in other problems like prediction of box office revenues, event detection and stock markets. One problem with social media data is that often many people choose to not share their posts with everyone but only with their friends which means data can sometimes underestimate the current activity level. For his experiment, he uses word count vectors and correlates them with load demand. "For the actual prediction, all occurring words in an hour are counted, then the algorithm searches the table containing the average load for a word and adds that load and current word count to a new table. We use the counts and words table to compute a new weighted average, which is the prediction for the number of hours ahead." We find this approach extremely simple and not very reliable.

Finally, (Kantardzic et al., 2015) advocate the use of social media data for improving models that were already trained on traditional data such as historical and weather data. While an implementation was not provided, they presented a "proof-of-concept that Web-based resources may give enough spatio-temporal information about human crowd in specific locations."

8.2 Candidate heterogeneous data sources

As previously mentioned, the factors that affect short-term load forecasting are 1) Date and time information e.g. hour of day, weekday or weekend, holidays etc. 2) Weather variables like temperature, humidity, wind speed and others 3) Human activity and behaviour patterns and 4) Economic factors like electricity price.

Traditional load forecasts rely mostly on historical data and in some cases include weather variables as well as calendar information. The data format of all these features is numeric. The same holds for economic factors. Considering the benefits of heterogeneous data that we covered in previous sections, it is interesting to investigate whether we can incorporate them for load forecasting. Some of the factors that affect load demand are captured with multiple modalities and this motivates us to pursue whether it is beneficial to include them. For example, weather information can be acquired not only from numerical measurements like temperature, humidity and so on, but also from Satellite and Whole Sky Images that depict cloud location and movement. Human activity can also be modelled using various text-based data sources like social media as well as mobile phone location data and GPS.

8.3 Weather factors

The weather variables that we usually consider for load demand forecasting are temperature, humidity, precipitation, wind speed and cloud cover. For all variables except cloud cover, there are no available data sources that describe them in a form other than numeric. However, cloud cover is an exception since it can be derived from images as well.

8.3.1 Cloud cover from Satellite and Whole Sky Images

Satellite images from space as well as Whole Sky Images that are taken from the ground contain a lot of information for the location and movement of clouds. The use cases of both these sources are plenty and also include Photovoltaic (PV) power forecasting which is a topic closely related to load forecasting since both play an important role in the function of the Smart Grid.

Satellite images

Satellite images of earth are an extremely valuable data source for the prediction of cloud location and movement. There are over 4500 satellites currently orbiting the planet with over 600 of them being imagery satellites (Rodziewicz, 2018). The best resolution currently available for public use, allows for each pixel of the image to cover an area of 25cm squared. Governments have access to higher resolution images but these are unavailable unless security clearance can be granted. There are both free and commercial sources for the acquisition of satellite images with the latter usually providing better resolution and frequency. We mentioned two of the three main properties of satellite images namely spatial resolution and temporal resolution which is how often we get an image of an area. The third one is spectral resolution which is a measure of their ability to resolve features in the electromagnetic spectrum. As humans, our eyes allow us to detect only certain wavelengths and when we think about an image, we usually think of the red, green and blue layers that compose it. However, the sensors in satellites can detect a lot more that we can, with some images having more than 12 layers with each layer providing more information about what is going on.

Whole Sky Images

Whole Sky Imagers (WSIs) are special cameras that are used for taking photographs of the sky from the ground up. They use a fish-eye lens in order to capture very wide areas and are useful for the prediction of cloud movement just like satellite images. WSIs can be used for short term forecast of cloud movement from real time to 5-30 minutes ahead by extracting cloud motion information from sky images. The process consists of four steps: 1) acquisition of sky images using devices like Whole Sky Imagers 2) identification of clouds by analysis of the images 3) cloud motion vector estimation from successive images and 4) use of cloud motion and location data for irradiance, cloud cover and solar power forecasting. Satellite image approaches are conceptually similar to WSI methods but offer certain advantages like larger spatial scales of cloud patterns and higher availability of satellite images.

Use of Satellite images and WSIs in PV forecasting

Photovoltaic (PV) farms are multiplying rapidly and the power generated by them is a significant fraction of the total power in the smart power grid. The generated power output of PV farms fluctuates according to meteorological conditions and accurate prediction of these fluctuations is important so that solar power trading on the energy market is made efficiently. (Pelland et al., 2013) compare different methods of PV forecasting and state that satellite images are commonly considered the best approach for forecasting up to 5 hours ahead. Persistence forecasts rely on historical power output of solar panels to predict future generation and ignore exogenous inputs. Statistical learning techniques are used to identify patterns in the time series data and estimate future power output. However, performance decreases when cloudiness is highly variable which justifies the use of satellite and whole sky images. Reference (Jang et al., 2016) utilizes a Support Vector Machine learning scheme to predict future irradiance and cloud amount for multiple areas in South Korea in a 15-300 minute range (intra-day), which can be utilized by an energy management system (EMS) for better grid operation. They extract Atmospheric Motion Vectors (AVM) from meteorological satellite images and combine them with cloud analysis information as well as irradiance images. AVMs capture wind direction and atmospheric motion information for upper, middle and lower wind fields. Cloud analysis images contain information about cloud thickness, shape and amount which are influential factors for the prediction of future cloud conditions. Finally, irradiance images show the amount of light that is reflected by the ground.

8.3.2 Potential value of satellite images for load forecasts

For the acquisition of cloud cover information, satellite and whole sky images provide a lot more spatial resolution than the cloud cover that is reported by weather stations. The latter is usually provided in the form of the fraction of the sky that is covered with clouds and therefore has a lot lower spatial resolution than satellite imagery. However, for the purpose of load forecasting in an urban environment, this extra resolution isn't necessary because the small differences in cloudiness in certain areas of the city and their effects in load demand are negligible when we consider the effects on the city as a whole. Moreover, the effect of cloud cover in electrical load demand is marginal when compared to other weather variables like temperature, humidity and wind speed (Janicki, 2017). Considering the reasons mentioned above, we believe there isn't enough potential benefit in incorporating satellite and whole sky images for the purpose of city-scale load forecasting.

8.4 Human activity factor

Inclusion of the human factor for load forecasting at an urban scale is a challenging task. Obtaining a completely accurate understanding of human dynamics is borderline impossible due to the complex factors that dictate every individual's decisions and actions. Our best bet is to approximate it with different indicators that reflect the mass behaviour of the area's population as a whole. Most load demand forecasts ignore the human activity factor and the few that do not, seem to model it in very simple ways. Studies that use the number of twitter posts indicate a correlation between the volume of posts and electricity needs (Luna et al., 2017) (Deng et al., 2018). Analysis of the textual content of posts to derive the human activity factor is still at an early stage. Correlation of words in social media posts to load is a way too simple approach and experimentation from (Stavast, 2014) supports this claim. Inclusion of the factor of big events was done by (Ding et al., 2013) using special calendars but there was no distinguishment of how big an event is relative to others. There is a lot of potential benefit in getting better insights of human activity through social media and other types of data since it is a very influential factor in load demand.

8.4.1 Detecting human activity in real time

The use of smartphones has become an extremely common thing in our era, making each and every one of us a sensor that collects and transmits data. Data that is generated from smartphones often includes location information from the user at the time the data was transmitted. Since people consistently use these devices on a daily basis, there is an enormous amount of data being generated every day. The abundance of available information makes it possible to detect events at the time of their happening and even obtain a population model of urban environments. Mobile devices can be used to contribute anonymously in order to obtain a real-time population view for a given location. A street-level granularity of human density is possible with analysis of data sources like mobile phone data, social media, GPS and other sensors. If we exploit all the available information that is present, we can acquire insights into the dynamic nature of mass human behaviour in urban environments.

Knowing the mobility patterns of people and how they change through time can provide existing load forecasting models with information about one of the most contributing factors to energy demand, which is human behaviour.

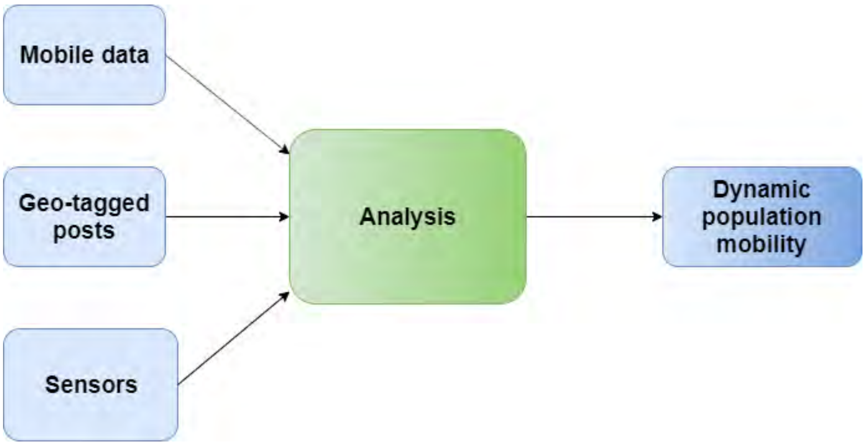

FIGURE 8.4: Proposed location data based event detection module.

Event detection and dynamic population estimation

Detection of social events in real time has been studied extensively (Zaharieva, Del Fabro, and Zeppelzauer, 2015) (Hu, Wang, and Li, 2018) (Liu and Jansson, 2018) using mainly social media data. A simple approach is given by (Garcia-Gasulla et al., 2014) where they only used social network activity and ignored the posts' contents to detect ongoing events. They first obtain a threshold of social network activity that corresponds to normal conditions and recognise any higher activity than that as an event. Even though it is a rare occurrence, natural disasters can alter load demand significantly and real time detection of them is possible with social media as (Athanasis et al., 2018) argues. Another useful data source is mobile phone call records which have been used as a proxy of human density in different areas of a city in (Selvarajoo, Schlapfer, and Tan, 2019) to obtain a dynamic population model within the city limits. (Cloquet and Blondel, 2014) use anonymous mobile phone call data to infer the flux of people heading towards an event which could be used as a way to predict attendance. Monitoring the traffic of the area is also insightful into load demand because the times at which people arrive at their homes where electricity is consumed, may be delayed due to heavy traffic (Aparicio et al., 2014). This can be done with various sensing devices and also with the analysis of social media and mobile data. (Goh, Koh, and Zhang, 2019) uses Vehicle Loop Detectors, mobile phone data as well as tweets to estimate crowd flows within the city. Similarly, (Zhang, Zheng, and Qi, 2017) predicts crowd flows for all areas in a city using trajectories from GPS and bike rent data. Another indicator of urban human activity is the amount of social network usage within the city's borders. (Luna et al., 2017) uses the volume of twitter posts as a predictor of load demand and a correlation between the two was observed. Even though the predictions were not very accurate, the volume of posts was the only variable they used and its inclusion in an already good feature set is potentially going to increase accuracy. Social media can also be used as a proxy of real-time human activity. (Liu et al., 2015) argues that estimation of the distribution of population in a country as well as its mobility is possible with geo-tagged tweets.

Research shows the potential of modeling the changes in population, in various temporal and spatial scales by analysing data that is generated from smartphones. There are various ways to acquire high-resolution time and location information of people in their daily lives such as mobile call records, geo-tagged social media posts and even WiFi access points as shown in (Sapiezynski et al., 2015). Each of these acquisition mediums has its advantages and drawbacks but they have shown great results in modeling human location and mobility, as shown in different studies. Mobile communication companies keep records of their customers' activity through Call Data Records (CDR). These records keep track of the time, location and other information every time a subscriber makes or receives a phone call or text message and for other technical reasons (Peters-Anders et al., 2017). An enormous amount of these data are generated on a daily basis which motivates researchers to explore their potential. As argued by multiple studies (De Meersman et al., 2016) (Selvarajoo, Schlapfer, and Tan, 2019), mobile data are effective for modeling population dynamics in various temporal and spatial scales ranging from country level (Deville et al., 2014) all the way down to urban scale (Khodabandelou et al., 2015) (Hu et al., 2009) (Li, Zhang, and Chen, 2019) (Dan and He, 2010), which is what we are interested in.

Another valuable data source for deriving dynamic population information is geo-tagged posts from social media like Twitter. Multiple studies use tweets as a proxy of human mobility with promising results (Patel et al., 2017). Analysis of population density maps derived from Twitter were shown to reflect human activities throughout the day (Tsou et al., 2018). We observe that there are currently many ways to derive a representation of human dynamics at an urban scale and judging by the increasing use of smartphones, the accuracy of these models will get better and better.

Dynamic population encoding

Our goal is to somehow include the dynamic human behaviour and mobility patterns in our forecasting models as an additional input feature. Our proposed approach includes dynamic population maps in numerical matrix form that represent human density in different regions of the area we are interested in. We begin by partitioning the area in a HxW grid, with each cell having the same size. For each cell, and consequently for each entry in the matrix, we calculate population density using data from mobile devices. The dimensions of the grid can be adjusted to match our desired granularity. This process is repeated for different points in time and the result is a time series of matrices that describe the dynamic population of the urban environment. An example of the grid partitioning is shown in (Fig. 8.5) and the respective matrix is depicted in (Fig. 8.6). Red marks represent detected people. Areas with a high concentration of red marks correspond to high values in the dynamic population matrix and areas with small crowds correspond to low values.

This matrix is essentially an image because it has spatial characteristics. This means that we can apply data fusion techniques in order to include it in the feature set of our machine learning models. We can complement the dynamic population features with information from event schedules. Events are known to attract human crowds and even though we are often unable to estimate the exact amount of attendance, we can potentially distinguish between large and small events through analysis of social media data.

8.4.2 Predicting human activity in advance

For forecasting events, the focus of most research is on civil unrest events and disease outbreaks, with the main focus of the prediction being the time of the event

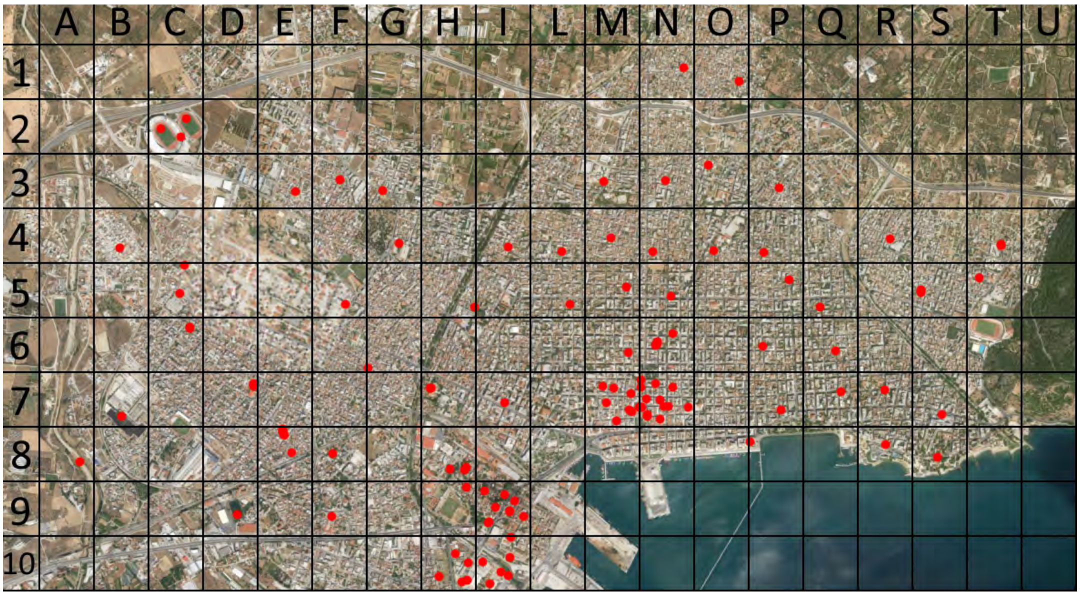

FIGURE 8.5: Dynamic population estimation through smartphone data. Background picture taken from Google maps.

(Zhao et al., 2015), while others include the spatial factor as well (Zhao et al., 2017a). That is not however the type of event we are interested in. Big social events like concerts and sport events are a factor for increases in electricity demand due to the large crowds they attract. Posts about social gatherings can provide valuable information about the use of electrical energy since such events attract large crowds from places outside the location where these events take place. Therefore, with more people in the area, there is a higher electrical energy demand. We require relatively good spatial accuracy for these events and current research on event forecasting from unstructured data doesn't seem to provide this, mainly because of insufficient location information in social media data (Zhao et al., 2017b). For automated identification of events with the spatial accuracy we need, our best option, in our opinion, is to extract them from sources that contain event schedules. It is also beneficial to distinguish between how big an event is, meaning how many people will attend it, in order to infer its impact on electricity demand. Event attendance was studied in (Xu et al., 2019) but only for events taken from an event-focused social network where people are invited to events and the prediction focused on an individual's decision to attend or not. Current research that incorporates big events for load forecasting only uses schedule information from special calendars and local websites (Ding et al., 2013) , (Kantardzic et al., 2015) to include them. It would be beneficial to approximate the public's interest in a certain event by mining social media data where people often post about events that they are excited about. Event popularity has not been studied for social gatherings, which is our interest in this thesis, but rather news events like stock market crashes, retirements of popular athletes, political events and similar topics (Chen, Kong, and Mao, 2017) , (Zhang et al., 2019). The prediction algorithms vary in complexity, from simple ones that are focused on the identification of "hot words" in posts and counting their frequency, to very complex approaches where the influence of related events is taken into account (Chen et al., 2018). It is possible in our opinion, to shift the focus of different event popularity techniques from their current events, which are not social-gathering-related, to the ones we are interested

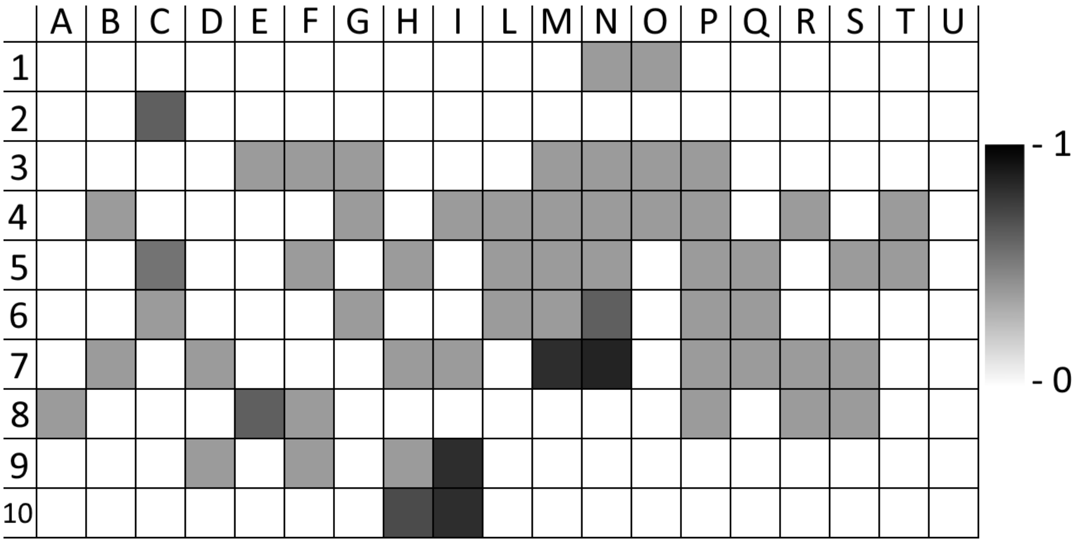

FIGURE 8.6: Corresponding dynamic population matrix from grid partitioning in (Fig. 8.5).

in such as concerts, sports events and others which attract large human crowds. Facebook event data is another valuable data source for predicting human activity. (Mezei, Pinter, and Felde, 2016) used Facebook event data to compute the popularity of an event as a weighted combination of users who selected attending, might attend and will not attend. This popularity is used for determining the congestion in different parts of the city in order to complement smart navigation systems for optimal route selection.

Event detection module

While we lack exact implementation details, we propose the development of an event detection module that predicts human activity intensity for a given future point in time using Web-based resources (Fig. 8.7). For example, if there is going to be a concert of a famous artist happening in the short future which is going to cause a lot of people to migrate to the location of the event, there are probably going to be posts about it in local web sites, blogs and in social media. The system should be able to detect this event and estimate the amount of people that are going to turn up. The output can be a measure of predicted human activity intensity for different points in time and should clearly be higher on special events. This information can be used by our machine learning models as an additional feature and fused with other sources of data, using the techniques that we analysed in previous chapters.

Despite our initial hopes of applying data fusion techniques with text data for load forecasting, there doesn't seem to be a reason to justify this action. In studies where text was fused with other types of data (Srivastava and Salakhutdinov, 2012) (Miech, Laptev, and Sivic, 2018), this was justified because all the multiple modalities that were used, comprised different descriptions of the same object. In our case, we want to use text data as a proxy of human crowd density so we focus on aggregated amounts of texts and in a sense, metadata of the original data. In our opinion, the context in which we use text data for load forecasting doesn't justify its

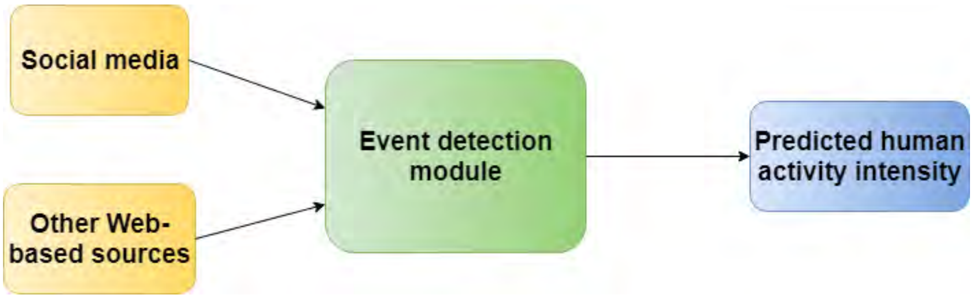

FIGURE 8.7: Proposed social media based event detection module.

fusion with other data types. Therefore, our machine learning models will accept a pre-processed version of the original text data which will be in numeric form.

Scheduled events encoding

We want to somehow encode spatial location information about the event as well. Inspired from other studies, we intend the output of the event detection module to be a matrix that describes the grid-partitioning of the area we are interested in predicting the load demand for. Our version uses Facebook data and specifically event information from the "Events" section.

The event module is responsible for building the event matrix which is going to be an additional input feature to our neural network architecture. We choose this data structure as an intuitive way to encode the spatio-temporal locations of social events as well as their expected attendance. This matrix is generated for each time-step with a frequency that matches the common sampling frequency of all other data sources, which is decided at the pre-processing stage. It has a similar structure to the dynamic population matrix and reflects the grid-partitioning of the city in blocks of identical size. During periods when no event is scheduled, the matrix has zeros in all of its entries. Each non-zero entry of the matrix at time step i signifies the existence of a social event during time i, at the area that corresponds to that position in the matrix. The value of that entry is a measure of the expected attendance/popularity of the event (Fig. 8.8).

Popularity measure

There are potentially many different ways to quantify popularity of events from social media. As a first step, we will use data from Facebook Events. Our main goal is to distinguish the size of different events in terms of how many people are likely to attend, since it is an indicator of the impact of that event in load demand. A concert of a local, relatively less known band isn't going to have the same effect as a concert of a worldwide known artist. Large-scale social events that have the potential to alter load demand significantly, are detectable very easily through social networks like Facebook. The "Events" section of Facebook is the most common way that we have personally been informed about local events and we believe the same holds for most people worldwide. This makes the problem of automatically detecting social events very easy to solve and computationally cheap. Users have the option to say whether they are interested in the event and if they are going to show up through the buttons "Interested" and "Going" respectively. The number of individual clicks is publicly available and even though there is usually some error compared to the

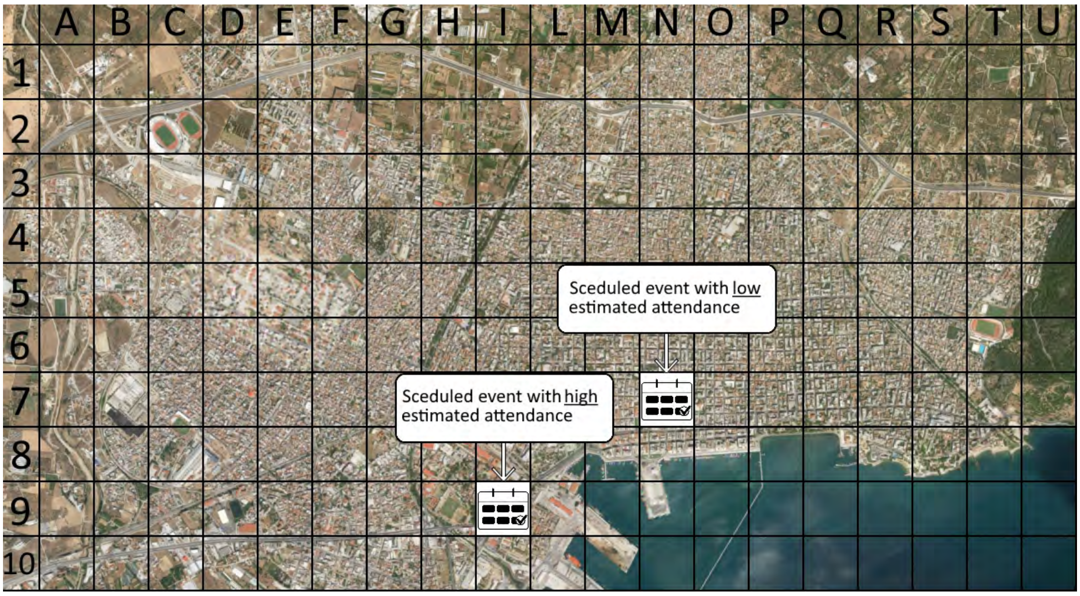
A
B
C
D
E
F
G
H
I
L
M
N
O
P
Q
R
S
T
U
1
2
3
4
5
6
7
8
9
10
Sceduled event with low estimated attendance
Sceduled event with high estimated attendance

FIGURE 8.8: Two scheduled events with different expected attendances at a given time.

actual counts, they are still good measures to distinguish the popularity of certain events compared to others. The final measure is going to be a weighted combination of the number of interested users and the number of users that confirmed that they are going, with the latter having a larger weight compared to the former e.g. 0.8 for "Going" and 0.2 for "Interested". The time of extraction of this data is also important. If we extract them very early, for example near the time of the announcement of the event, the numbers will probably underestimate the attendance since many users wouldn't have been informed about the event yet or haven't made up their minds. The extraction time should be close to the beginning of the event when counts of expected attendees are more stable, which gives us more confidence that the numbers represent the actual popularity of the event and are less prone to changes.

We observe that we can either predict human activity a long time in advance through social media analysis or detect it in real time through various heterogeneous data sources like mobile data, GPS, sensors and social media. Proper combination of both methods is very promising in terms of predicting human activity more accurately. The dynamic population information that is encoded in the dynamic population matrix can be complemented with the special event information that we encode in the event matrix. Through proper combination of both these matrices, we provide the system with more knowledge about the state of urban population at each time step. This information can be used by load forecasting models as an additional feature and in terms, provide more accurate forecasts since we incorporate more contextual information about the determining factors of load demand to them.

8.5 Load forecasting with dynamic population and event schedule information

Our proposed approach incorporates information from the majority of factors that affect electricity load demand. This includes time and calendar information, weather

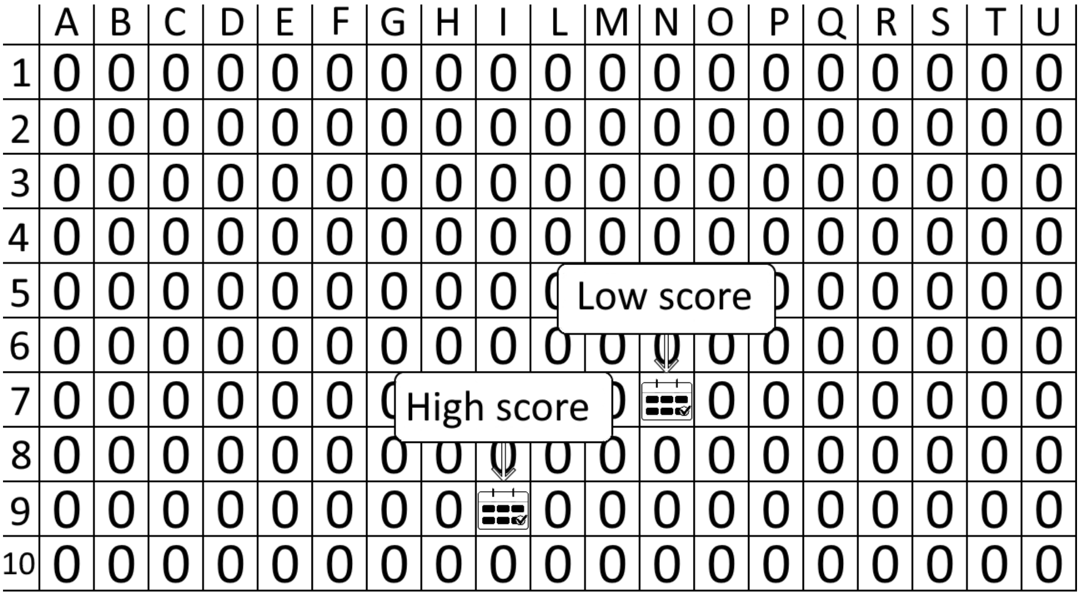

FIGURE 8.9: Corresponding event matrix from grid partitioning of (Fig. 8.8).

factors, real-time electricity prices and human activity information. Previous load demand measurements will also be included in the feature set, in the form of lagged values. Despite not conducting an experiment ourselves, we will theoretically go through the process of STLF if we had access to the necessary data. Our scenario will be for hourly load forecasting, even though the time frame can be adjusted for shorter on longer forecasts.

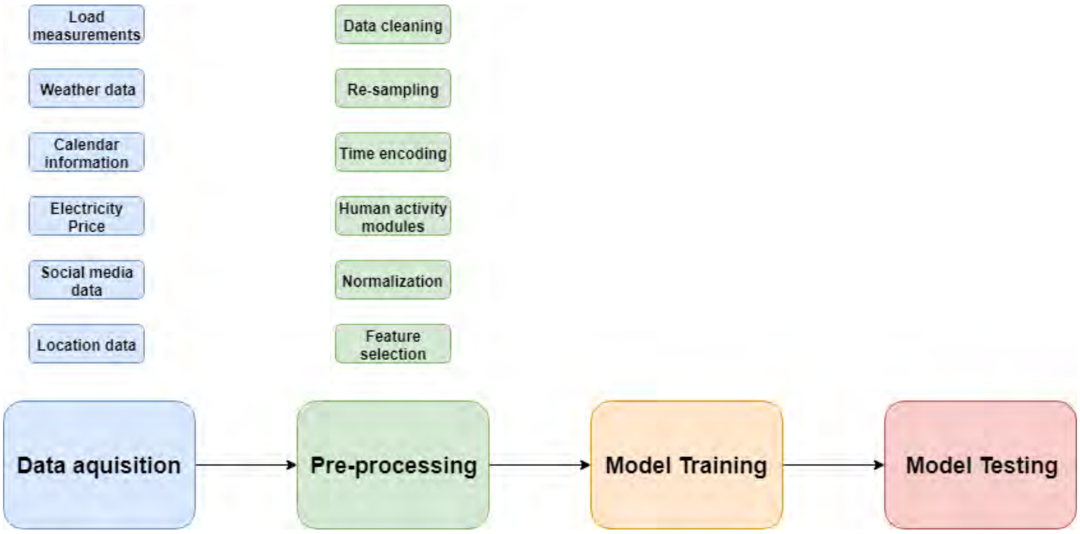

FIGURE 8.10: Proposed load forecasting framework.

8.5.1 Data Collection

For historical load measurements, data acquisition is rather easy. If the dataset consists of non-aggregated data of individual meter measurements for individual consumers, we need to sum the measurements of unique users and arrive at the desired aggregated load consumption measurements. Weather data is available through weather stations which will provide temperature, wind speed, precipitation, humidity and cloud cover data. Calendar data is trivial to collect although it is important to include special days e.g. local holidays so that proper encoding of them can be made in the pre-processing stage. While many economic factors like income, type of customers, GDP growth etc are influential to load demand, their effects are negligible for short-term forecasting. Electricity price however, is influential for the short-term as well because it varies during the day and datasets for price are publicly available. For the inclusion of the human activity factor, we require calculation of the dynamic population and event schedule matrices. We don't claim expertise for this task, therefore we suppose that they are already provided through analysis of social media and location data as mentioned in previous sections.

8.5.2 Data Pre-processing

The most tedious step is pre-processing and it is hard to know beforehand all the data preparation stages we must apply. However, some of the most important ones are:

Data cleaning

To deal with missing values and outliers, we need to first inspect their nature. If we have data Missing Completely At Random (MCAR), we can perform simple imputation with the mean, mode or median of the variable but if this is not the case, we have to use other techniques like the ones mentioned in 3.1.

Re-sampling

Another common issue with using multiple data sources is that measurement frequency can vary from one to the other. Before building our forecasting models, we require all the different data measurements to be in time-series format with the same sampling rate. We need to choose the desired common sampling rate while taking into consideration potential negative effects of re-sampling certain variables. After we arrive at the final sampling rate, we match all variables to this rate with either up-sampling or down-sampling. The properties of each variable dictate the optimal interpolation or aggregating strategy we should apply. For example, when down-sampling load measurements from two hour to one hour intervals, we have to sum nearby values but when doing the same for weather variables, averaging or interpolation techniques are required.

Time encoding

Calendar information has to be properly encoded and new features need to be created for effective modeling of the temporal changes of electricity demand. Categorical variables like "day of the week", "weekend or weekday", "month", "season" and "special day" need to be generated from the calendar. We will use binary one-hot encoding with dummy variables for categorical variables with more than two categories like "month" which has 12 distinct values. This will obviously increase the number of features.

Normalization

Because different variables have different scales, this can cause issues to machine learning models like favoring certain variables with large values and ignoring variables with relatively small values. Decreases in learning times and other negative impacts are also to be expected. To combat this, all variables need to be brought to the same scale of 0 to 1.

Correlation analysis and feature selection

Even though we believe our variables are influential to load demand, it is a good practice to first perform correlation analysis in order to distinguish features that are more important than others and perhaps discard the ones under a certain threshold. Selecting an optimal sub-set of the original features can increase performance and prevent the model of being "confused" by uncorrelated features. If we want our model to include past values of load demand as predictors, we must decide which past values are most correlated with the present load demand. The auto-correlation function can be used for this purpose. We are particularly fond of the approach of (Tong et al., 2018) where for load forecasting of day k, measurements of the previous day $(k-1)$ were used in combination with measurements from the past 3 weeks $(k-7, k-14, k-21)$. For our case which is hourly forecasting, the time lags differ but the main intuition remains the same.

8.5.3 Model Training

Since the human activity factor is included in the form of matrices with spatial characteristics, we treat them as images. They have to be fused with multi modal data fusion techniques like multiple kernel learning, deep learning with sub-networks for each modality or ensembles. All these techniques and possibly many more are applicable in our case. Experimentation with different algorithms and model architectures is necessary in order to find what approach is optimal in terms of performance and computational complexity. For our demonstration, we choose the deep learning route and assign different sub-networks for the image data, which consists of the two human activity matrices, and numeric data which is the type of all other factors we take into account. There are two different ways that we can combine the dynamic population and event schedule matrices (Fig. 8.11). The first is to assign separate Convolutional Neural Networks (CNN) to each matrix and concatenate their extracted features with the rest of our inputs which include weather variables, calendar information, electricity price and lagged values of electricity demand. The second way is to use each matrix as a separate channel in the CNN branch, in the same manner as images are traditionally fed into CNNs with 3 channels namely R, G and B. The rest of the architecture remains the same. Experiments with both approaches as well as completely different methods and combinations of features should be conducted to conclude which approach is best.

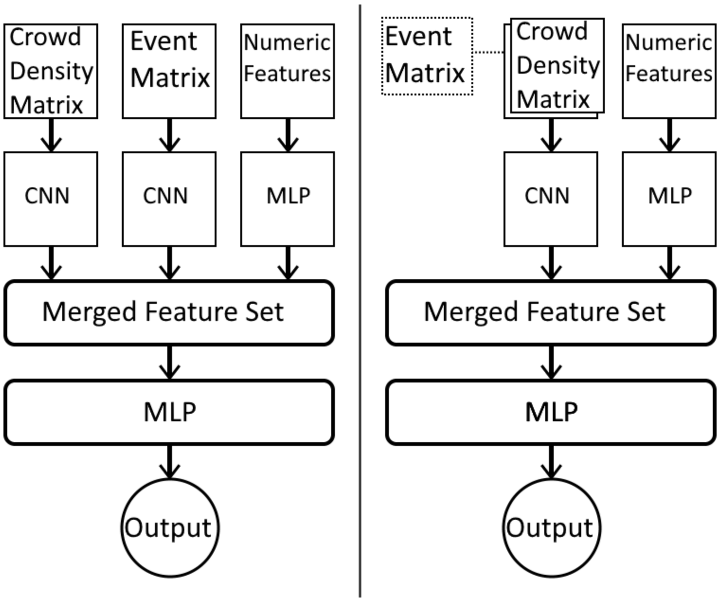

FIGURE 8.11: Described architectures.

8.5.4 Model Testing

Different error measures will be used to validate our models and decide which is the best in terms of performance. Some of the most frequently used for load forecasting are the Mean Absolute Percentage Error (MAPE) and Mean Square Error (MSE). We have had issues in the past with MAPE in the presence of zero values but this is not likely here because load demand is always greater than zero.

8.6 Conclusion

In our opinion, short term load forecasting can be improved with the inclusion of features that describe human behaviours and mobility. The enrichment of the feature set with more contextual information about the factors that contribute to energy demand can lead to more accurate forecasts. Our proposed approach relies on using dynamic population and event schedule maps in the form of matrices as additional features to our forecasting models. These matrices have spatial characteristics because they represent a geographical area and we therefore treat them as images. This means that we have to use multimodal data fusion techniques in order to combine them with the rest of our features, which are all in numeric form. The fusion can be performed with different methods, including the ones we reviewed in early chapters, such as kernel methods, deep learning and ensembles.

Chapter 9

Discussion

9.1 Synopsis

Heterogeneity is one of the most common aspects of big data since the majority of the total generated data is heterogeneous. There are many different types of heterogeneity but in general, we usually refer to data as being heterogeneous when it originates from different sources, modalities or scales. Multi-modality is our main interest and it refers to the acquisition of information about a target object through different modalities such as images, text and normal numerical data. We are interested in fusing data that comes from multiple data sources, mainly because it contains more information than single-source data since it provides us with different aspects of our target object. Due to several negative characteristics that are associated with them, heterogeneous data pose a challenge to traditional machine learning techniques and often require several pre-processing stages before their fusion, which we briefly looked at after our introduction. We distinguished data fusion techniques based on the stage of the machine learning process that the combination takes place. Early fusion combines the information at the feature level while late fusion does it at the decision level. Both approaches have their advantages and drawbacks. During our review, we came across different algorithms and methods that work with heterogeneous data, with the most common ones being kernel methods, deep learning and ensembles. From the kernel methods we encountered, Multiple Kernel Learning stood out as an intuitive way to perform data fusion by finding an optimal mix of kernels, where each is created using a different data source. Deep learning proved to be just as effective in data fusion, due to its feature extraction capabilities and the flexibility it provides us when designing architectures. Similarly to MKL for kernel methods, the technique that stood out to us in deep learning, is to assign different multi-layer sub-networks to different modalities and merge their output layers in a common feature vector. Ensembles allow us to use different models for each of our multiple data sources and combine their predictions. The obvious advantage, besides the large flexibility they provide, is the ability to exploit the superiority of certain models when working with certain data types. After mentioning some of the methods that we encountered less often during our study, we investigate the potential of using heterogeneous data for short-term load forecasting. Our peer review on STLF showed little attention by researchers on this topic. Considering the most influential factors of electricity demand in the short term, we found weather and human activity to be the only candidates for heterogeneous data fusion, since we can acquire information for them using multi-modal data sources. For weather data, the idea was to complement the forecasted numerical weather variables with satellite and whole sky images that provide high spatial accuracy of cloud locations. In our opinion, this wouldn't be very helpful for forecasting load demand more accurately, although they are very useful to photovoltaic solar power forecasting which plays an important role in the proper operation of the smart grid. Therefore, we focused on ways to include the human activity factor, which is highly variable and complex. Our proposition is to use social media and mobile data for the extraction of event schedule information as well as dynamic human population estimation for the urban location of interest. We encode both pieces of information in matrices that represent the geography of the area, partitioned in a grid. Due to their spatial characteristics, we treat those matrices as images and therefore apply multi-modal data fusion techniques to include them as features to our models.

9.2 Research and development prospects

In our research we mainly focus on using human activity data in conjunction to traditional variables that are used in short term load forecasting. However there are some problems related to collecting all the necessary data. The main problem is that the required mobile data may not be easily accessible. Call details records are not available to the public and are only used by law enforcement agencies. If somehow electric companies are provided access to such data, the privacy aspect needs to be handled. To address privacy concerns, data anonymization techniques need to be applied to CDR datasets in order to encrypt information that can be used for personal identification of users. Moreover the collecting of GPS data may be problematic. The general public may not want to sacrifice their privacy and allow GPS data to be collected. For this reason, an incentives program may help acquire this data and reward citizens for allowing their data to be collected.

Furthermore, an actual implementation of the proposed technique is necessary to verify the validity of the model and discover the benefits of integrating human activity data for the purpose of STLF. Since population density is constantly changing, the incorporation of this information is probably only beneficial for forecasting load in the very-short term. We cannot be certain about the exact optimal forecast horizon, although we expect that population information will be helpful for forecasting from minutes to a few hours ahead. Different forecast horizons need to be evaluated in order to find what is optimal and practical.

Evaluation of which type of location data (CDR, geo-tagged posts, GPS) is applicable for a given area is also important. For example, even though geo-tagged social media posts have shown potential in modelling human mobility, this ability depends on the geographical area that the experiments take place. Areas where a small fraction of the population uses social media will not provide enough spatio-temporal information to our model. If we intend to use social media data, large and densely populated cities with high social media activity should be the areas where the experiments need to take place first.

As far as the event schedule information is concerned, we need to evaluate whether including it in our models is beneficial or not. Data acquisition for events is not a challenging task, at least for the simple method that we proposed which uses Facebook Event data. This can be done with a simple API that collects event information for a particular area.

Future technological advancements like the adaptation of 5G networks is promising in terms of improving the population mapping abilities. Location data from CDRs is derived from the nearest cell tower that the user is connected. With 5G networks, base stations will be more densely placed in geographical space, providing higher spatial accuracy of the users.

Considering that the proposed technique works, it can be used to provide new insights on how human mobility patterns affect energy demand. Through the analysis of dynamic population maps and the electricity demand that they correspond to, we can discover relationships between the two that we didn't know about. This can potentially help in more efficient planning of the electricity grid.

Finally, an important factor for load demand to look out for in the future, is the electrification of vehicles and their increasing popularity worldwide. There is a continuous growth in the electric vehicle (EV) market. This growth will lead in a future where the electric vehicles outnumber the internal combustion engine vehicles. This future is not far and the integration of EVs in energy prediction may play an important role in load forecasting.

Bibliography

Aparicio, Juan et al. (2014). "Exploiting road traffic data for Very short term load forecasting in Smart Grids". In: *2014 IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2014*, pp. 1–5. DOI: 10.1109/ISGT.2014.6816498.

Arandjelovic, Relja et al. (2018). "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6, pp. 1437–1451. ISSN: 01628828. DOI: 10.1109/TPAMI.2017.2711011. arXiv: arXiv:1511.07247v3.

Arputhamary, B. and L. Arockiam (2015). "Data Integration in Big Data Environment". In: *Bonfring International Journal of Data Mining* 5.1, pp. 01–05. ISSN: 2250107X. DOI: 10.9756/bijdm.8001.

Athanasis, Nikos et al. (2018). "The emergence of social media for natural disasters management: A big data perspective". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 42.3W4, pp. 75–82. ISSN: 16821750. DOI: 10.5194/isprs-archives-XLII-3-W4-75-2018.

Audebert, Nicolas, Bertrand Le Saux, and Sébastien Lefèvrey (2017). "Fusion of heterogeneous data in convolutional networks for urban semantic labeling". In: *2017 Joint Urban Remote Sensing Event, JURSE 2017*. DOI: 10.1109/JURSE.2017.7924566. arXiv: arXiv:1701.05818v1.

Aziz, Mohammad S. and Chandan K. Reddy (2010). "Robust prediction from multiple heterogeneous data sources with partial information". In: p. 1857. DOI: 10.1145/1871437.1871747.

Ballard, Chris and Wenjia Wang (2016). "Dynamic ensemble selection methods for heterogeneous data mining". In: *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)* 2016-Septe, pp. 1021–1026. DOI: 10.1109/WCICA.2016.7578244.

Baltrusaitis, Tadas, Chaitanya Ahuja, and Louis Philippe Morency (2019). "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2, pp. 423–443. ISSN: 19393539. DOI: 10.1109/TPAMI.2018.2798607. arXiv: arXiv:1705.09406v2.

Breckels, Lisa M. et al. (2016). "Learning from Heterogeneous Data Sources: An Application in Spatial Proteomics". In: *PLoS Computational Biology* 12.5, pp. 1–26. ISSN: 15537358. DOI: 10.1371/journal.pcbi.1004920.

Chatterjee, Moitreya et al. (2015). "Combining two perspectives on classifying multimodal data for recognizing speaker traits". In: *ICMI 2015 - Proceedings of the 2015 ACM International Conference on Multimodal Interaction*, pp. 7–14. DOI: 10.1145/2818346.2820747.

Chen, Guandan, Qingchao Kong, and Wenji Mao (2017). "An attention-based neural popularity prediction model for social media events". In: *2017 IEEE International Conference on Intelligence and Security Informatics: Security and Big Data, ISI 2017*, pp. 161–163. DOI: 10.1109/ISI.2017.8004898.

Chen, Guandan et al. (2018). "A partition and interaction combined model for social event popularity prediction". In: *2018 IEEE International Conference on Intelligence and Security Informatics, ISI 2018*, pp. 232–237. DOI: 10.1109/ISI.2018.8587366.

Chen, Quanjun et al. (2016). "Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference". In: *Aaai*, pp. 338–344.

Cheng, Yao et al. (2019). "PowerNet: Neural Power Demand Forecasting in Smart Grid". In: arXiv: 1904.11979. URL: http://arxiv.org/abs/1904.11979.

Chung, Seungeun et al. (2019). "Sensor data acquisition and multimodal sensor fusion for human activity recognition using deep learning". In: *Sensors (Switzerland)* 19.7. ISSN: 14248220. DOI: 10.3390/s19071716.

Cloquet, C. and V.D. Blondel (2014). "Forecasting event attendance with anonymized mobile phone data". In: *Big Data Research*, pp. 1–26.

Dan, Yu Fang and Zhongshi He (2010). "A dynamic model for urban population density estimation using mobile phone location data". In: *Proceedings of the 2010 5th IEEE Conference on Industrial Electronics and Applications, ICIEA 2010*, pp. 1429–1433. DOI: 10.1109/ICIEA.2010.5514844.

De Meersman, Freddy et al. (2016). "Assessing the quality of mobile phone data as a source of statistics". In: *European Conference on Quality in Official Statistics* June, pp. 1–16.

Deng, Chengbin et al. (2018). "Social media data as a proxy for hourly fine-scale electric power consumption estimation". In: *Environment and Planning A* 50.8, pp. 1553–1557. ISSN: 14723409. DOI: 10.1177/0308518X18786250.

Deville, Pierre et al. (2014). "Dynamic population mapping using mobile phone data". In: *Proceedings of the National Academy of Sciences of the United States of America* 111.45, pp. 15888–15893. ISSN: 10916490. DOI: 10.1073/pnas.1408439111.

Ding, Yong et al. (2013). "A framework for short-term activity-aware load forecasting". In: *ACM International Conference Proceeding Series*, pp. 23–28. DOI: 10.1145/2516911.2516919.

Filkov, Vladimir and Steven Skiena (2010). "Heterogeneous Data Integration with the Consensus Clustering Formalism". In: pp. 110–123. DOI: 10.1007/978-3-540-24745-6_8.

Garcia-Gasulla, Dario et al. (2014). "Social network data analysis for event detection". In: *Frontiers in Artificial Intelligence and Applications* 263, pp. 1009–1010. ISSN: 09226389. DOI: 10.3233/978-1-61499-419-0-1009.

Ghassemi, Marzyeh et al. (2015). "A Multivariate Timeseries Modeling Approach to Severity of Illness Assessment and Forecasting in ICU with Sparse, Heterogeneous Clinical Data." In: *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence* 2015, pp. 446–453. ISSN: 2159-5399. URL: http://www.ncbi.nlm.nih.gov/pubmed/27182460{\%}0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4864016.

Goh, Gary, Jing Yu Koh, and Yue Zhang (2019). "Twitter-informed crowd flow prediction". In: *IEEE International Conference on Data Mining Workshops, ICDMW* 2018-Novem, pp. 624–631. ISSN: 23759259. DOI: 10.1109/ICDMW.2018.00097.

Gönen, Mehmet (2014). "Embedding heterogeneous data by preserving multiple kernels". In: *Frontiers in Artificial Intelligence and Applications* 263.i, pp. 381–386. ISSN: 09226389. DOI: 10.3233/978-1-61499-419-0-381.

Guo, Kehua et al. (2019). "iFusion: Towards efficient intelligence fusion for deep learning from real-time and heterogeneous data". In: *Information Fusion* 51.July 2018, pp. 215–223. ISSN: 15662535. DOI: 10.1016/j.inffus.2019.02.008. URL: https://doi.org/10.1016/j.inffus.2019.02.008.

Gupta, Vikas (2017). "An Overview of Different Types of Load Forecasting Methods and the Factors Affecting the Load Forecasting". In: *International Journal for Research in Applied Science and Engineering Technology* V.IV, pp. 729–733. DOI: 10.22214/ijraset.2017.4132.

Haghighat, Mohammad, Mohamed Abdel-Mottaleb, and Wadee Alhalabi (2016). “Discriminant Correlation Analysis: Real-Time Feature Level Fusion for Multimodal Biometric Recognition”. In: *IEEE Transactions on Information Forensics and Security*. ISSN: 15566013. DOI: 10.1109/TIFS.2016.2569061.

He, Jingrui (2017). “Learning from data heterogeneity: Algorithms and applications”. In: *IJCAI International Joint Conference on Artificial Intelligence*, pp. 5126–5130. ISSN: 10450823.

Hu, Jinxing et al. (2009). “Dynamic modeling of urban population travel behavior based on data fusion of mobile phone positioning data and FCD”. In: *2009 17th International Conference on Geoinformatics, Geoinformatics 2009*, pp. 1–5. DOI: 10.1109/GEOINFORMATICS.2009.5293222.

Hu, Jun, Yuxin Wang, and Ping Li (2018). “Online city-scale hyper-local event detection via analysis of social media and human mobility”. In: *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017* 2018-Janua, pp. 626–635. DOI: 10.1109/BigData.2017.8257978.

Jang, Han Seung et al. (2016). “Solar Power Prediction Based on Satellite Images and Support Vector Machine”. In: *IEEE Transactions on Sustainable Energy* 7.3, pp. 1255–1263. ISSN: 19493029. DOI: 10.1109/TSTE.2016.2535466.

Janicki, Marcin (2017). “Methods of weather variables introduction into short-term electric load forecasting models - a review”. In: *Przeglad Elektrotechniczny* 93.4, pp. 70–73. ISSN: 00332097. DOI: 10.15199/48.2017.04.18.

Kantardzic, Mehmed et al. (2015). “Improved Short Term Energy Load Forecasting Using Web-Based Social Networks”. In: *Social Networking* 04.04, pp. 119–131. ISSN: 2169-3285. DOI: 10.4236/sn.2015.44014.

Khodabandelou, Ghazaleh et al. (2015). “IEEE TRANSACTIONS ON MOBILE COMPUTING 1 Estimation of Static and Dynamic Urban Populations with Mobile Network Metadata”. In: arXiv: 1810.12909v1. URL: https://arxiv.org/pdf/1810.12909.pdf.

Korkmaz, Gizem et al. (2015). “Combining Heterogeneous Data Sources for Civil Unrest Forecasting”. In: pp. 258–265. DOI: 10.1145/2808797.2808847.

Lahat, Dana, Tulay Adali, and Christian Jutten (2015). “Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects”. In: *Proceedings of the IEEE* 103.9, pp. 1449–1477. ISSN: 00189219. DOI: 10.1109/JPROC.2015.2460697.

Lanckriet, Gert (2007). “Sparse and large-scale learning with heterogeneous data”. In: *youtube.com*.

Lewis, Darrin P., Tony Jebara, and William Stafford Noble (2006). “Support vector machine learning from heterogeneous data: An empirical analysis using protein sequence and structure”. In: *Bioinformatics* 22.22, pp. 2753–2760. ISSN: 13674803. DOI: 10.1093/bioinformatics/btl475.

Li, Mingxiao, Hengcai Zhang, and Jie Chen (2019). “Fine-Grained Dynamic Population Mapping Method Based on Large-Scale Sparse Mobile Phone Data”. In: Mdm, pp. 473–478. DOI: 10.1109/mdm.2019.00008.

Liu, Jiajun et al. (2015). “Multi-scale population and mobility estimation with geo-Tagged Tweets”. In: *Proceedings - International Conference on Data Engineering* 2015-June, pp. 83–86. ISSN: 10844627. DOI: 10.1109/ICDEW.2015.7129551.

Liu, Shuhua and Patrick Jansson (2018). “City event detection from social media with neural embeddings and topic model visualization”. In: *Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017* 2018-Janua.2012, pp. 4111–4116. DOI: 10.1109/BigData.2017.8258430.

Long, Kejun et al. (2018). “Predicting Freeway Travel Time Using Multiple- Source Heterogeneous Data Integration”. In: *Applied Sciences* 9.1, p. 104. DOI: 10.3390/app9010104.

Luna, Ana et al. (2017). “Power demand forecasting through social network activity and artificial neural networks”. In: *Proceedings of the 2016 IEEE ANDESCON, ANDESCON 2016*. DOI: 10.1109/ANDESCON.2016.7836248.

Ma, Chunmei et al. (2016). “Representation Learning from Time Labelled Heterogeneous Data for Mobile Crowdsensing”. In: *Mobile Information Systems* 2016, pp. 1–10. ISSN: 1574-017X. DOI: 10.1155/2016/2097243.

Mezei, Miklos, Gergo Pinter, and Imre Felde (2016). “Urban mobility by Facebook events”. In: *INES 2016 - 20th Jubilee IEEE International Conference on Intelligent Engineering Systems, Proceedings* Figure 2, pp. 223–226. DOI: 10.1109/INES.2016.7555124.

Miech, Antoine, Ivan Laptev, and Josef Sivic (2018). “Learning a Text-Video Embedding from Incomplete and Heterogeneous Data”. In: arXiv: 1804.02516. URL: http://arxiv.org/abs/1804.02516.

Nikolopoulos, Vassilis (2012). “Data Fusion Theory for Smart Grids”. In: June, pp. 1–15.

Patel, Nirav N. et al. (2017). “Improving Large Area Population Mapping Using Geotweet Densities”. In: *Transactions in GIS* 21.2, pp. 317–331. ISSN: 14679671. DOI: 10.1111/tgis.12214.

Pelland, Sophie et al. (2013). “Photovoltaic and Solar Forecasting : State of the Art”. In: p. 40.

Peng, Yang et al. (2016). “Multimodal ensemble fusion for disambiguation and retrieval”. In: *IEEE Multimedia* 23.2, pp. 42–52. ISSN: 1070986X. DOI: 10.1109/MMUL.2016.26.

Peters-Anders, Jan et al. (2017). “Dynamic, interactive and visual analysis of population distribution and mobility dynamics in an urban environment using the mobility explorer framework”. In: *Information (Switzerland)* 8.2. ISSN: 20782489. DOI: 10.3390/info8020056.

Ramesh, Bharathi (2019). “MACHINE LEARNING ALGORITHMS FOR HETEROGENEOUS DATA : A COMPARATIVE”. In: 10.3, pp. 9–19.

Ray, Priyadip et al. (2014). *Bayesian Joint Analysis of Heterogeneous Data*. Tech. rep.

Rodziewicz, Damian (2018). *Deep learning in Satellite imagery*. URL: https://www.kdnuggets.com/2018/12/deep-learning-satellite-imagery.html (visited on 07/26/2019).

Sapiezynski, Piotr et al. (2015). “Tracking human mobility using WiFi signals”. In: *PLoS ONE* 10.7, pp. 1–11. ISSN: 19326203. DOI: 10.1371/journal.pone.0130824.

Selvarajoo, Stefan, Markus Schlapfer, and Rui Tan (2019). “Urban Electric Load Forecasting with Mobile Phone Location Data”. In: *2018 Asian Conference on Energy, Power and Transportation Electrification, ACEPT 2018* September, pp. 1–6. DOI: 10.1109/ACEPT.2018.8610757.

Society, Max Planck (2010). “Kernel Methods for Fusing Heterogeneous Data (PPT)”. In:

Sokolov, Artem et al. (2013). “Combining heterogeneous data sources for accurate functional annotation of proteins”. In: *BMC Bioinformatics* 14.SUPPL.3, S10. ISSN: 14712105. DOI: 10.1186/1471-2105-14-S3-S10. URL: http://www.biomedcentral.com/1471-2105/14/S3/S10.

Srivastava, Nitish and Ruslan Salakhutdinov (2012). “Learning Representations for Multimodal Data with Deep Belief Nets”. In: *Icml*, pp. 1–8. URL: https://pdfs.

semanticscholar.org/5555/b28607cada5474bca772e1cc553b624415c9.pdf{\%}0Afile:///Files/F7/F7260D68-548D-416E-9F4B-7D7BB77D7EEE.pdf.

Stavast, P (2014). “Prediction of Energy Consumption Using Historical Data and Twitter”. In: March.

Tong, Chao et al. (2018). “An efficient deep model for day-ahead electricity load forecasting with stacked denoising auto-encoders”. In: *Journal of Parallel and Distributed Computing* 117, pp. 267–273. ISSN: 07437315. DOI: 10.1016/j.jpdc.2017.06.007. URL: http://dx.doi.org/10.1016/j.jpdc.2017.06.007.

Tsou, Ming-Hsiang et al. (2018). “Estimating hourly population distribution change at high spatiotemporal resolution in urban areas using geo-tagged tweets, land use data, and dasymetric maps”. In: arXiv: 1810.06554. URL: http://arxiv.org/abs/1810.06554.

Wang, Binhuan et al. (2019). “Additive partially linear models for massive heterogeneous data”. In: *Stat* 8.1, pp. 391–431. ISSN: 20491573. DOI: 10.1002/sta4.223.

Wang, Yixing et al. (2018). “Short-term load forecasting with multi-source data using gated recurrent unit neural networks”. In: *Energies* 11.5. ISSN: 19961073. DOI: 10.3390/en11051138.

Wikipedia (2019). *Data fusion — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Data%20fusion&oldid=912923358. [Online; accessed 17-September-2019].

Xu, Tong et al. (2019). “Exploiting the dynamic mutual influence for predicting social event participation”. In: *IEEE Transactions on Knowledge and Data Engineering* 31.6, pp. 1122–1135. ISSN: 15582191. DOI: 10.1109/TKDE.2018.2851222.

Xu, Zenglin, Irwin King, and Michael R. Lyu (2007). “Web page classification with heterogeneous data fusion”. In: p. 1171. DOI: 10.1145/1242572.1242750.

Yann LeCun, Yoshua Bengio, Geoffrey Hinton (2015). “Deep learning (2015), Y. LeCun, Y. Bengio and G. Hinton”. In: *Nature*.

Zaharieva, Maia, Manfred Del Fabro, and Matthias Zeppelzauer (2015). “Cross-Platform Social Event Detection”. In: *IEEE Multimedia* 22.3, pp. 14–25. ISSN: 1070986X. DOI: 10.1109/MMUL.2015.31.

Zhang, Junbo, Yu Zheng, and Dekang Qi (2017). “Deep spatio-temporal residual networks for citywide crowd flows prediction”. In: *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 1655–1661. arXiv: arXiv:1610.00081v2.

Zhang, Lili et al. (2018). “Multi-source heterogeneous data fusion”. In: *2018 International Conference on Artificial Intelligence and Big Data, ICAIBD 2018*, pp. 47–51. DOI: 10.1109/ICAIBD.2018.8396165.

Zhang, Qidong et al. (2019). “Adaptive General Event Popularity Analysis on Streaming Data”. In: *Proceedings - 5th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, BDCAT 2018*, pp. 116–125. DOI: 10.1109/BDCAT.2018.00022.

Zhao, Lei, Qinghua Hu, and Wenwu Wang (2015). “Heterogeneous Feature Selection with Multi-Modal Deep Neural Networks and Sparse Group LASSO”. In: *IEEE Transactions on Multimedia* 17.11, pp. 1936–1948. ISSN: 15209210. DOI: 10.1109/TMM.2015.2477058.

Zhao, Liang et al. (2015). “Spatiotemporal event forecasting in social media”. In: *SIAM International Conference on Data Mining 2015, SDM 2015*, pp. 963–971.

Zhao, Liang et al. (2017a). “Feature Constrained Multi-Task Learning Models for Spatiotemporal Event Forecasting”. In: *IEEE Transactions on Knowledge and Data Engineering* 29.5, pp. 1059–1072. ISSN: 10414347. DOI: 10.1109/TKDE.2017.2657624.

Zhao, Liang et al. (2017b). “Multi-resolution spatial event forecasting in social media”. In: *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 689–698. ISSN: 15504786. DOI: 10.1109/ICDM.2016.75.

Zhou, Chongyu et al. (2017). “Learning Deep Representations from Heterogeneous Patient Data for Predictive Diagnosis”. In: pp. 115–123. DOI: 10.1145/3107411.3107433.