Hybrid Model for Interpretable Time Series Analysis

A THESIS

Presented to the Department of Computer Science and Computer Engineering

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

Option in Computer Science

Committee Members:

Ju Cheol Moon, Ph.D

Roman Tankelevich, Ph.D

Seok-Chul Kwon, Ph.D

College Designee:

Hamid Rahai, Ph.D.

By Ruben Rosales

May 2020

Abstract

In this thesis, we study interpretable machine learning as applied to complex-valued time-series data. Scientists have studied the use of several machine learning methods such as Convolutional Neural Networks, Recurrent Neural Networks, and Support Vector Machines for time series classification. These methods, however, fall short of allowing users to visualize patterns within their dataset.

To address this issue, we propose an interpretable hybrid model that can be extended to any time series dataset. In the majority of existing work in the field of interpretability, black-box models tend to outperform white-box models consistently. However, instead of relying purely on one method, we propose a collaboration between the two. This gives us the performance of a black-box while allowing us to visualize features of the majority of our dataset.

To accomplish this, we created a framework composed of two models, an ensemble method of classifiers that functions as a black-box method, and a white-box model that encodes time series as images. The white-box model attempts to classify them through a neural network and outputs all images correctly classified in both black-box and white-box models.

Acknowledgments

Contents

Illustrations

Figures

Tables

Chapter 1

Introduction

As time series datasets become ever so popular the need for deep learning models becomes greater. In our black-box model, we analyze deep learning models such as Convolutional Neural Networks which have become increasingly popular and widely used in the last decade but their complex nature makes it difficult for users to understand what is going on inside [5]. So, as black-box models get more popular and increase in complexity the need for an interpretable accessory is needed so we propose a hybrid model consisting of a black-box for classification and an interpretable white-box model which highlights characteristics of a time series for users to understand.

The goal of our work is not to find an alternative to black-box methods but to provide a way for users to have an understanding of what features are prevalent in their dataset.

Contributions

We focus on creating deep learning architectures to classify our datasets as well as extending existing work to create our white-box model. Our final results show how robust our black-box method is as well as support for our white-box model.

Chapter 2

Data

We analyzed our model on three radio signal datasets generated using Spatial Modulation. Spatial Modulation is a transmission technique that uses multiple antennas. It maps a block of information bits to two units, one is a symbol chosen from a constellation diagram, and the second one is a unique transmit antenna number that is chosen from a set of transmit antennas [3]. The method in which transmit antennas send and receive radio waves can be described through polarization. Two popular basis polarizations are horizontal linear, H, and vertical linear, V. Our datasets contain data horizontally transmitted and vertically received (HV) and data vertically transmitted and vertically received.

The two properties HV and VV, are given to us in raw discrete format. Both properties consist of time and amplitude values but vary in length and in how they are processed. The raw signal consists of over 31,000 data points with the exact time it was received. The size of the discrete data can vary from 22 to 44 data points and is a processed representation of the raw signal that has equally spaced values. In this thesis we focus on discrete data given that it is closer to datasets in real-world situations.

Chapter 3

Related Work

Chapter 4

Preprocessing

Radio signal data is presented in a complex-valued format that is unusable in a typical neural network, so in an attempt to make our model robust and reduce any complexity, we focused on using real values as features.

We tried numerous methods to extract real-valued features, including Fourier Transform, Short Time Fourier Transform, a custom sliding window method, and polar coordinates taken directly from the signals amplitude/phase value. Before applying any of these methods, we normalized all data using L2 normalization.

Transformations

Fourier Transform

The Fourier Transform (FT) is a mathematical tool that decomposes any function into a sum of sinusoidal basis functions. Each basis function is a complex value of a frequency, so it allows us to view our data in the frequency domain as opposed to amplitude.

One of the shortcomings of the FT is rooted in the Heisenberg Uncertainty Principle (HUP) [1]. The HUP states that the position and velocity of an object cannot be simultaneously measured, which can be applied to the time-frequency information of a signal. This means we cannot know which spectral components exist at any given time. The closest we can get is sampling at different ranges of time and finding a range of frequencies within that time frame. This method is described as the Short-Time Fourier Transform.

Short-Time Fourier Transform

The Short-Time Fourier transform (STFT) is a Fourier related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. [1] Computing STF requires the signal to be divided into segments of equal length and then have the FT applied to that segment. This allows us to view the Fourier spectrum at a more granular level, which could potentially reveal different patterns amongst signals of different classes.

One of the issues with this method is that we can create very narrow window sizes, which gives us a better understanding of the data with respect to time, but we lose understanding of the frequency domain. Additionally, selecting an appropriate window size for segmenting the signal can be an arduous task that would require fine-tuning as well as increasing the dimensionality of our dataset.

Sliding Window Method

We propose a method similar to STFT but to reduce the size of the data we take the mean at each window which would turn our size into size N where N is equal to the number of windows we use. This performed as well as STFT and gave us the advantage of reducing the dimensionality of our data.

Polar Coordinates

The representation of a complex number as a sum of a real and imaginary number, $z = x + iy$, is called its Cartesian representation. For every cartesian point, we calculate its radius and angular distance which are real values and use those as features. This method

outperformed the previous three methods but limited our feature size which made it difficult to get consistent accuracy across all three of our datasets.

Wavelets

Wavelets are similar to the FT in which they deconstruct a signal using representations of other signals. The key difference is that wavelet signals are finite in time and frequency as opposed to sin and cos signals, which can carry on forever [4]. This allows us to extract information from a signal with respect to time and location.

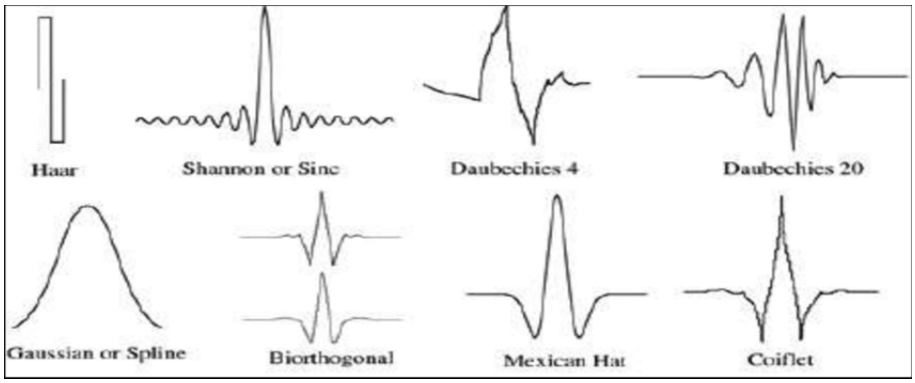

Figure 4.1: Example of different types of wavelets.

The use of wavelets is called the wavelet transform, which is a technique in which a signal is analyzed using different versions of a dilated and translated basis functions called the mother wavelet. There are two types of wavelet transformations, discrete and continuous. In this thesis, we focus on discrete. Discrete Wavelet Transform uses a discrete set of wavelet scales and translations which decompose the signal into a mutually orthogonal set of wavelets.

We take advantage of wavelets by applying discrete wavelet transforms as a filterbank, which means it is composed of cascading high-pass and low-pass filters. This allows

us to split a signal into several frequency sub-bands. We focus on wavelet decomposition as our feature extractor because it outperformed all other methods in terms of accuracy and training time.

Data Shape

A neural network requires all data to be the same shape. However, because we downsample in wavelet decomposition, each data size is halved until making it impossible to place all levels of decomposition into the same dataset. In order to circumvent this, we tried two different methods, resampling the data and treating each level of decomposition as its own dataset.

Resampling

We use spline interpolation in order to resize each level of decomposition into a single array. We tested different sizes of interpolation from N to N/3, where N is the size of the largest discrete signal size, and found no reduction in performance.

Each level

Figure 5.5 depicts our model in which each level of decomposition is treated as its own dataset. We found no difference in accuracy compared to the resampling method. However, given that related work we found treated each decomposition level as its own dataset, we decided to go with this method.

Chapter 5

Models

Definition

Machine learning algorithms can be seen as learning a target function (f) that maps input data (X) to an output (Y). There are several techniques to make this work, but we focus on nonparametric algorithms, namely Convolutional Neural Networks (CNN) and Long Short Term Memory networks (LSTM). Nonparametric algorithms are algorithms that attempt to make minimal assumptions about the form of the function so they can learn any form from data provided to it. An example would be a neural network because it has no prior knowledge about what it is classifying and attempts to generalize any new data points that it has not seen before.

CNN

A Convolutional Neural Network is a type of deep neural network that can be applied to different domains such as computer vision or time series analysis. It consists of an input and output layer as well as hidden layers. A convolution is an operation between a vector of weights w against an input x. It consists of taking the dot product between m and x were in steps of a filter size defined by n.

Convolutional Neural Networks perform feature learning via non-linear transformations implemented as a series of layers. The input data is a multidimensional array, called a tensor. The input data is passed through an input layer, followed by a series of hidden

layers to extract features, and finally, an output layer, which in the case of classification, gives a probability for each class.

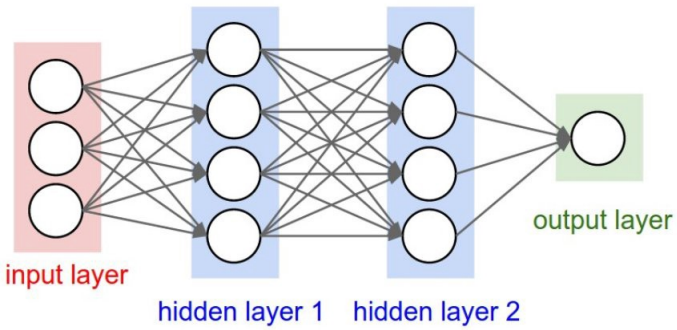

Figure 5.1: Example of a Convolutional Neural Network.

Hidden layers are crucial to neural networks because it is how a model can determine which data representations are useful for explaining the relationships in the given data. Each layer consists of several kernels that perform a convolution over the input; therefore, they referred to as convolutional layers. Kernels are feature detectors that convolve over the input and produce a transformed version of the data at the output.

LSTM

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points but also entire sequences of data.

A standard LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing, and making predictions based on time series data since there can be lags of unknown duration between significant events in a time series. LSTMs were developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs.

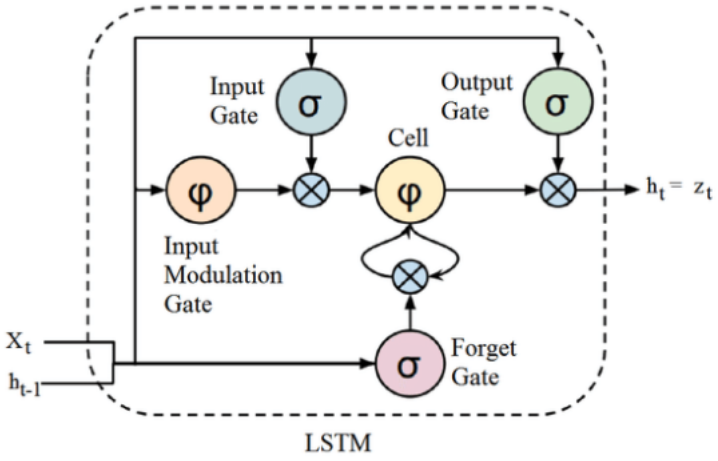

Figure 5.2: Example of a LSTM unit.

Multimodal Deep Learning

Due to the superior performance and computationally tractable representation capability in multiple domains such as visual, audio, and text, deep neural networks have gained tremendous popularity in multimodal learning tasks [2]. Typically, domain-specific neural networks are used on different modalities to generate their representations, and the individual representations are merged or aggregated. Finally, the prediction is made on top of aggregated representation, usually with another neural network to capture the interactions between modalities and learn complex function mapping between input and output.

Architectures Used

CNN

Our CNN architecture consists of 3 convolutional layers each followed by a batch normalization and dropout layer then finally connected to two dense layers (Figure X). We attempted to make it deeper but found inconsistent performance across all of our datasets.

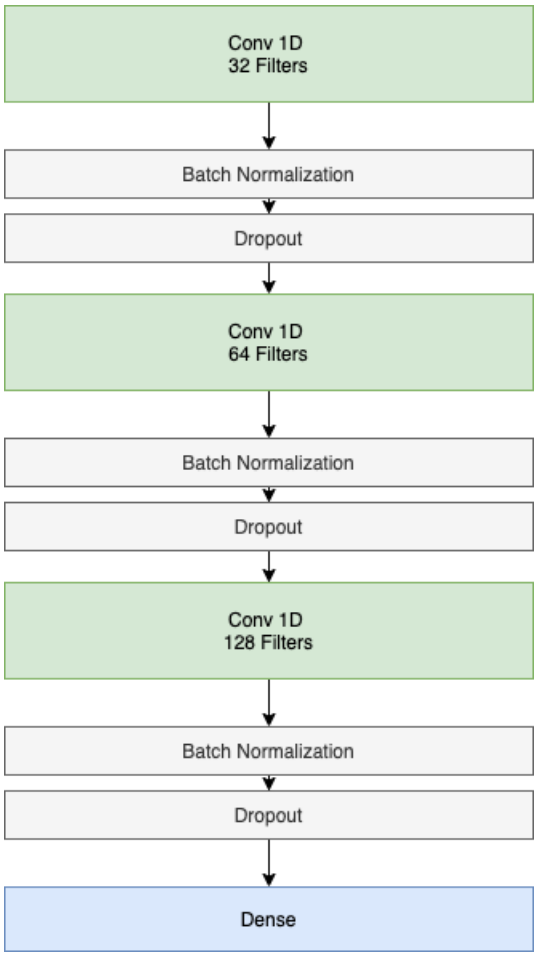

Figure 5.3: CNN architecture used in our black-box model.

LSTM

We wanted to keep our LSTM network as small as possible, for X purposes, so we went with two LSTM layers of 128 hidden states and a dropout rate of 4/10 followed by a

dense layer of 128 connections. We attempted different state sizes but found 128 to be the smallest number with the best consistent performance.

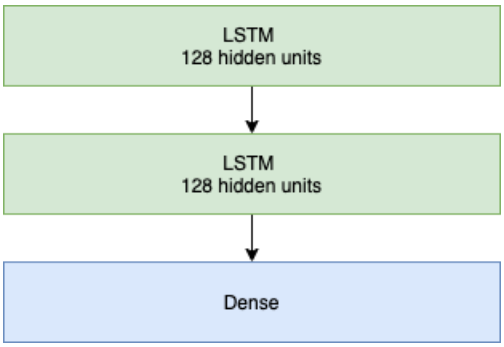

Figure 5.4: LSTM architecture used in our black-box model.

Final Model

We propose a multimodal learning architecture for the purpose of time series classification which is illustrated in Figure 5.5. The multimodal architecture allows our data to vary in size and given that all three levels of decomposition have varying length due to downsampling, we propose each model to learn the representation of each distinct feature. For this architecture we propose two architectures, a 1D CNN and an LSTM, in which we employ two CNN's and 1 LSTM. They are all concatenated after their respective dense layer after which they perform classication via a softmax layer.

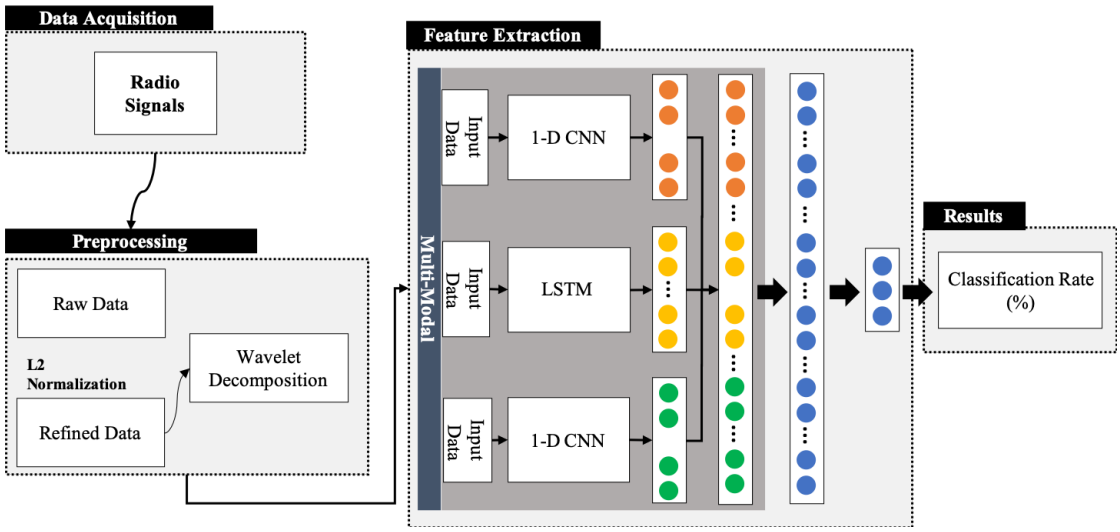

Figure 5.5: Overview of our black-box model.

Chapter 6

Markov Transition Fields

We propose a framework similar to [6] for encoding dynamical transition statistics, but we continue their work by i^{th} order Markov transition probabilities.

Given a time series X, we decompose its magnitude axis into three separate properties, P_1, P_2, P_3. We then identify the Q quantile bins for each property and assign each $x \in P_1, P_2, P_3$ to its corresponding bin q_j (j in [1, Q]). Thus, we construct three QxQ adjacency transition matrices, W_1, W_2, W_3, by counting transitions among quantile bins in the manner of a i^{th} order Markov chain along the time axis.

	A	B	C	D
A	0.917	0.083	0	0
B	0.083	0.583	0.334	0
C	0	0.260	0.522	0.218
D	0	0.083	0.167	0.75

Figure 6.1: Example of QxQ quantile bin matrix used to calculate MTF.

W does not take into account the temporal axis so to prevent any information loss we construct a Markov Transition Field, M, for each W. The MTF denotes the probability of transitioning from q_i to q_j for each $x \in P$. This, in turn, allows us to consider the transition probability on the magnitude and temporal axis.

$$M = \begin{bmatrix} W_{ij}|x_1 \in q_i, x_1 \in q_j & \dots & W_{ij}|x_1 \in q_i, x_n \in q_j \\ \vdots & \ddots & \\ W_{ij}|x_n \in q_i, x_1 \in q_j & \dots & W_{ij}|x_n \in q_i, x_n \in q_j \end{bmatrix}$$

Figure 6.2: Markov Transition Field where W_{ij} denotes the transition probability from quantile i to j.

As described in [6] the MTF encodes the multi-span transition probabilities of the time series, but given that we have three different M's we modify their approach and consider each M to be a separate color channel of RGB where M_1 is red, M_2 is blue, M_3 is green. Since each row in M is a probability from 0 to 1 we multiply it by 255 to get a color value.

Chapter 7

Main Results

Black-Box Model

Our black-box model is constructed of three different architectures, 1 LSTM and 2 CNN's, that are concatenated at the dense layer. The three features we use are amplitude/phase of each level of decomposition for HV and VV. We chose four because we wanted to take into consideration both HV / VV in the event that one had more descriptive features than the other.

To test our model we split our data into training and test sets by randomly selecting 70% of our data as test and left the other 30% as our test set. We then performed 3 fold cross validation and took the average of our results as our final outcome.

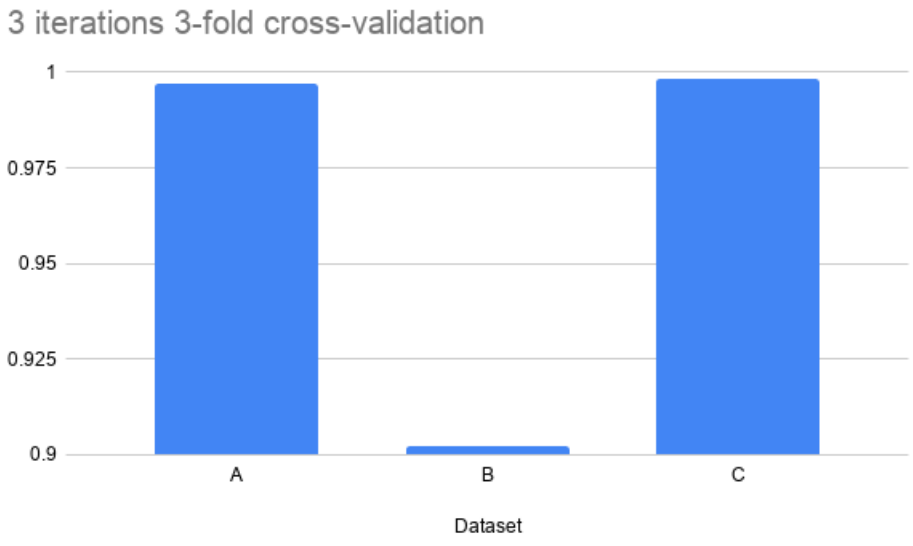

Figure 7.1: Results

Dataset B

Given that this dataset is ten times larget than our other two datasets we wanted to try different training and test splits, namely we went as low as 8% for training data and used the remaining 92% as test data. We ran the same number of cross validation as our previous tests and found minimal difference in performance.

White-Box Model

Our white box is composed of a model that takes in raw data as an input and converts each signal into an image through the MTF method. We then pass those images through a CNN and those that are correctly classified are shown to the user.

In order to classify those images we created a CNN for it specifically and avoided using our existing CNN in order to make it deeper and increase performance when images aren't that classifiable. We explored several architectures, such as, ResNet, Inception-v4, and AlexNet but we decided to create an architecture similar to AlexNet due to the variation and size of datasets we used we couldn't create a model deeper or more complex than it otherwise performance would drop because we didn't have enough data to properly train it.

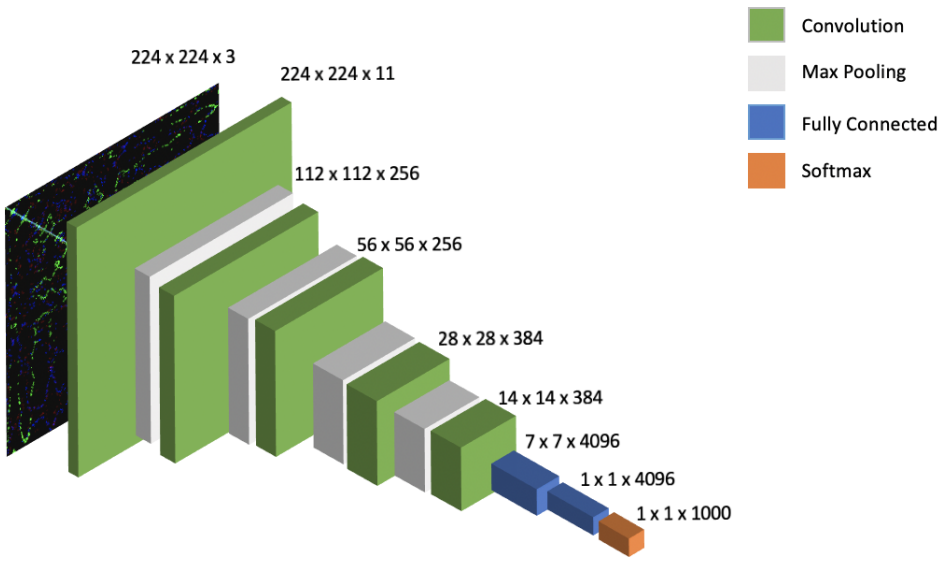

Figure 7.2: CNN architecture for classifying images generated through MTF's.

Chapter 8

Conclusion

Throughout this thesis, we have stressed the importance of a hybrid machine learning algorithm as well as providing a framework that can successfully classify and visualize different classes when applied to time-series data.

We proposed an extension to existing work described in [] so users can visualize their data while being able to classify them with 60% accuracy on our dataset and model. We also created a robust black-box model consisting of two different deep learning architectures that consistently provided competitive accuracy.

In the future, we would like to explore other methods for our white-box model that would act as an alternative to Markov Transition Fields in hopes that it produces higher accuracy than our current model. Additionally, we would like to extensively test our model on different time-series datasets, not just radio signal datasets.

Bibliography

[1] Mitch Hill. THE UNCERTAINTY PRINCIPLE FOR FOURIER TRANSFORMS ON THE REAL LINE. page 17.

[2] Edward Kim and Kathleen F. McCoy. Multimodal Deep Learning using Images and Text for Information Graphic Classification. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '18, pages 143–148, Galway, Ireland, October 2018. Association for Computing Machinery.

[3] Raed Y. Mesleh, Harald Haas, Sinan Sinanovic, Chang Wook Ahn, and Sangboh Yun. Spatial Modulation. *IEEE Transactions on Vehicular Technology*, 57(4):2228–2241, July 2008. Conference Name: IEEE Transactions on Vehicular Technology.

[4] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

[5] Tong Wang and Qihang Lin. Hybrid Predictive Model: When an Interpretable Model Collaborates with a Black-box Model. *arXiv:1905.04241 [cs, stat]*, May 2019. arXiv: 1905.04241.

[6] Zhiguang Wang and Tim Oates. Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks. page 7.