Project Design for Beat The Heat

Kaylee Pham, David Shin, Saulo Rubio, Sergio Ramirez
Tyler Poplawski, Macoto Ward, Lior Zlotikman*
{kaylee.pham.704,david.shin.904,saulo.rubio.491,sergio.ramirez.754}@my.csun.edu
{tyler.poplawski.456,david.ward.761,lior.zlotikman.183}@my.csun.edu
Department of Computer Science
California State University, Northridge
Northridge, California

1 INTRODUCTION

While much research has been conducted on how to predict wildfires with great accuracy, wildfires are still a major ongoing issue due to climate change. Between January and October 2020 wildfires burned 3.75 million acres, killed 29 people, and destroyed 8,169 structures in California [1]. Thus, a better model is still needed to predict the creation and spread of wildfires. The goal of this project is to create a reliable and accurate wildfire prediction model to help our community in solving this major problem. The project design begins with preparing and feeding the data to the neural network and support vector machines models. These models in turn predict the creation and spread of wildfire. The final part of the design measures the quality and accuracy of the prediction. The rest of the paper organized as follows: Section 2 provides an overview of the main components of the design; Section 3 will go further into details about the machine learning algorithms and processes used in the design.

2 DESIGN

As shown in Fig. 1, this project is composed of three major components: preparation, model building, and quality measurement. First part of preparation is gathering datasets from multiple data sources. Then extract only the needed data, and transform it into a new dataset that will be used in next component. For model building, the new dataset will be split into two sets for training and testing. Next, the training set will be fed into the model and validated after the training phase is completed. If the model is not properly validated, it will loop back to training until it passes the validation phase. Finally, test the model against the testing dataset. This process will be applied on the Neural Network (NN) model and the Support Vector Machine (SVM) model. Output the results from testing and send it to the last phase. In quality measurement, the prediction result will be observed and analyzed. Additionally, the quality measurement phase will reprocess the data if improvement is needed. Then compare NN and SVM's accuracy and evaluate them to choose a best model for deployment.

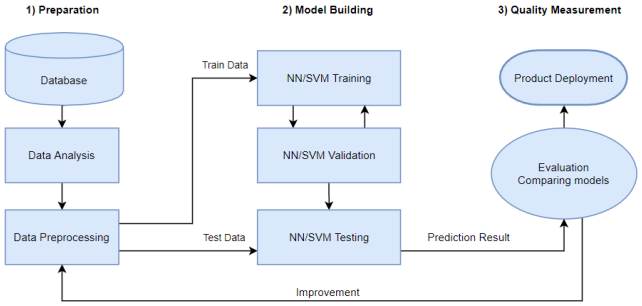

Figure 1: Design Overview

3 METHODOLOGIES

This section will address the processes and algorithms that will be used in the design. The first subsection covers the data preparation used to prepare and format the data to fit the models. The next two sections will discuss the algorithms, neural network and support vector machine, used to predict the creation and spread of wildfires. The last subsection will examine the quality measurement of the predictions.

3.1 Data Preparation

Figure 2 illustrates the process of preparing the data. The data is first extracted from the respective databases. The model uses two streams of data, one will be used to train and validate a machine learning model using SVM and NN (left), while the other will used with realtime data to be tested against a working model (right).

The databases used to build the training model (left two) come from a past fire record database and meteorological data from a satellite database, whereas the test model dataset (right) is from live remote sensing data from a satellite. The two data streams are nearly identical processes except that the live remote sensing data isn't integrated into another dataset and it isn't split into training and validation datasets.

*All authors contributed equally to this research.

COMP490 Senior Design Project, California State University, Northridge,

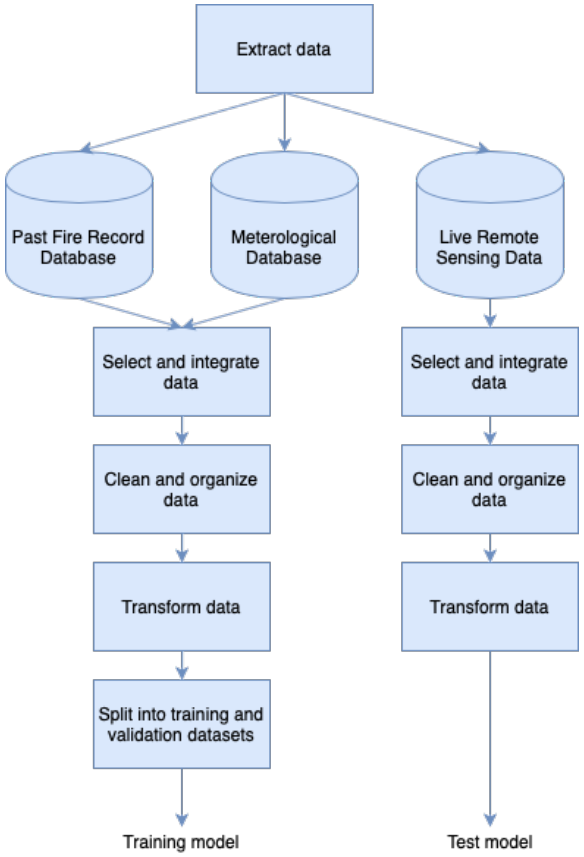

Figure 2: Data pipeline overview
Pipeline of data starting from the database phase showing two streams of data, one for training and validating a model (left), and the other for feeding live data to a model to increase accuracy (right). Steps are nearly identical for both processes but some steps are omitted for the live data stream. Steps include selecting data, integrating datasets, cleaning data, organizing data, and transforming it before feeding it to their respective models. The training dataset divides the data into training and validation sets for the model.

3.1.1 Select and integrate. Variables in the datasets are analyzed based on their influence on the target output, spatial spread, and those above a measure of correlation are chosen over other variables with less influence on the target output. The prioritized meteorological and past fire record variables are then integrated into a singular dataset.

3.1.2 Clean and organize. After the high priority variables are chosen, it is time to clean and organize the data to best fit into a format that will be easy to feed to a training model. First the data is formatted into a format best suited for the chosen models (e.g. relational database format, csv, json, etc.). Cleaning the data refers to removing or fixing datapoints that are incomplete or removing information that doesn't pertain to the target output such as satellite data that very far from the target area. Sampling is a common technique used to reduce the size of the data in order to focus on the high priority areas in order to reduce the computational load in the later processing steps.

3.1.3 Transform data. Transforming data or feature engineering is the last step of packaging the data for the model and includes scaling, decomposition, and aggregation. Scaling variables is the process of unifying a scale across all variables such that they all have a consistent scale for measurement for the machine learning methods. This could look like transforming a quantity such as relative humidity into a scale between 0 and 1. Decomposition is the separation of complex variables into simpler subunits to improve the ML methods ability to discern the most important information. For example, from a time and date variable, a decomposed version might produce a time of day variable which correlates better with temperature and humidity than just a time variable. Aggregation is the inverse of decomposition where less impactful variables are combined to produce a measure that is more useful to the problem at hand.

3.1.4 Split into training and validation datasets. The left data stream is then split into two different sets, one to train the model and one to be used against the model for validation.

3.2 Neural Network

The neural network component to be used will follow a feed forward neural network style and make use of the backpropagation algorithm. "Feed forward" refers to a non-cyclical style of neural network, in which the data is passed through multiple layers of neurons strictly from input to output without looping back. Backpropagation is used to help identify which neurons contribute to errors in the output from the network. When errors are encountered, the weights between connections are modified and the model is retrained to correctly map its inputs to outputs and resolve these errors [2].

Referring to Fig. 3, the following steps of the neural network flow can be traced. First, the training set of data is passed into the input neurons and initial weights are chosen. This data is fed forward and a linear combination of the inputs and weights is calculated, which becomes the input for the hidden layer(s). From this layer the data passes through an activation function, typically a sigmoid function is used, and compared against a threshold value to determine if that neuron's output is passed on. Linear combination is again calculated before arriving at the output layer. Here is where error checking takes place, and the backpropagation algorithm occurs if weights need to be modified. Finally, this entire process is repeated until the best fitting weights are discovered and the model's accuracy is maximally improved.

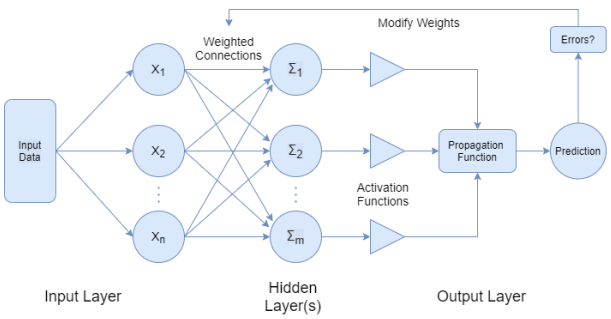

Figure 3: Neural network workflow

3.3 Support Vector Machine

Support Vector Machines (SVM) are commonly used for classification and regression problems. The main objective of an SVM is to find the optimal hyperplane for the classification of two classes. An easy way to explain how an SVM works is by using a 2D space representation of two different classes linearly separable. The optimal hyperplane is the line passing between these two classes, creating a maximum margin that serves as the host of the support vectors. Unfortunately, not all datasets can be linearly separable; in a multidimensional space, Kernel functions are used to shape the hyperplane. To find the optimal hyperplane in a multidimensional space, a Support Vector Machine(SVM) has the following major components: Fig. 4

- **Feature Selection**: helps create new features so that the Kernel function can transform and find boundaries in the dataset.
- **SVM Kernel**: functions used to shape the hyperplane in a multidimensional space.
- **Classifier (SVM)**: finds the optimal hyperplane and the support vectors that define the maximum margin.

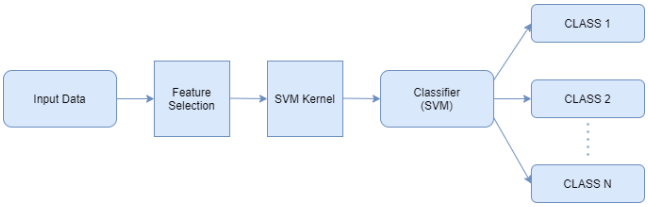

Figure 4: Support Vector Machine

3.3.1 *SVM Kernel.*

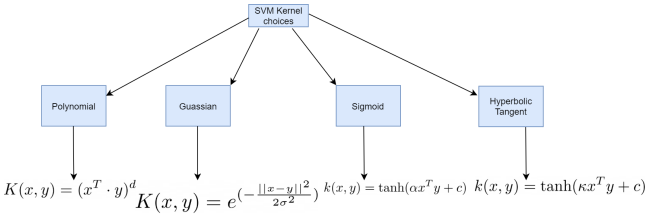

Figure 5: SVM Kernels

For the SVM algorithm we choose a kernel, a map from often single dimensional space to n-dimensional space. This is done in the hopes that a higher dimensional space will allow the data to be classified more efficiently. And as previously mentioned in the section above, being able to find a plane that will linearly separate the data is an extremely integral part of the SVM model. For our data and design we found that the following 4 kernels are potential candidates: polynomial, Gaussian, Sigmoid, and Hyperbolic Tangent. In figure 4 above we can see that for each given kernel, we have an associated function. And interestingly, choosing any such kernel will allow us to make predictions based only on our original feature selections.

We found that the polynomial kernel is often associated with having high measure of accuracy in the realm of of image processing. The Gaussian kernel is looked at as being more general purpose. That is to say that it primarily operates without being given prior knowledge of input data. Moreover, the remaining two kernels, Sigmoid and Hyerbolic Tangent, form an excellent pair with neural networks. What we hope to accomplish by selecting a kernel model, is to choose one such model that will help us deliver the most accurate and reliable results.

3.4 Quality Measurement

The Quality Measurement phase is the last phase in the design and the work flow can be observed in Fig. 6. In this phase, we will be collecting the prediction results from our trained models, applying evaluation methods, comparing and finalizing our results.

The results from the prediction phase will be collected and the accuracy will be calculated for both the Neural Network and Support Vector Machine Model. The accuracy can be measured in one of two ways. It can be measured through the summation of True Positive and True Negative predictions, divided by the total summation of True Positive, True Negative, False Positive and False Negative predictions. Alternatively, the accuracy can be measured by using a statistical model (RMSE, MSE, MAE, etc).

- **True Positive** - actual and predicted outcomes are the true.
- **True Negative** - actual and predicated outcomes are false.
- **False Positive** - actual outcome is true, but predicted to be false.
- **False Negative** - actual outcome is false, but predicted to be true.

The statistical models calculate the average distance of error between the predicted and the actual outcome. After the accuracy has been calculated, we will compare the results.

The accuracy of each model be compared to each other as well as the scores from previous works to see if adjustments are needed.

If adjustments should be made, we will go back into the Data Preparation Phase or more specifically, the Data Preprocessing Phase, as shown in Figure 6, to make adjustments to parameters. The adjustments will be making are either to change the weight of the parameters or to replace them. After we have reached the desired results, we will finalize them by writing our report on our process and findings.

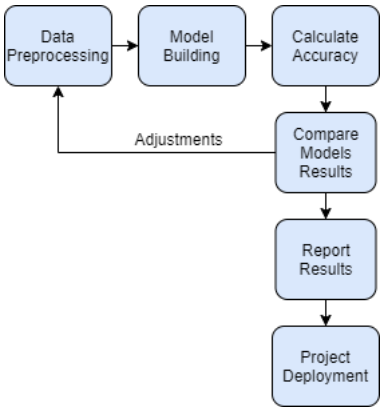

Figure 6: Quality Measurement Design which shows the work flow for this phase

ACKNOWLEDGMENTS

We would like to thank our professor, Dr. Xunfei Jiang for all the support and insight she has given us in the development of this project.

REFERENCES

[1] 2020 (accessed October 1, 2020). *2020 Incident Archive.* https://www.fire.ca.gov/incidents/2020/

[2] J. Amrutha and A. S. Remya Ajai. 2018. Performance analysis of backpropagation algorithm of artificial neural networks in verilog. *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2018 - Proceedings* (2018), 1547–1550. https://doi.org/10.1109/RTEICT42901.2018.9012614