Wildfire Prediction and Reduction for the West Coast of the USA Using a Neural Network Approach

SShamtej Singh Rana

Commack High School

Abstract

In the first ten months of 2019, on the west coast of the United States, more than 40,000 wildfires burned about 3.2 million acres of forest, costing over $900 million in damages, in addition to 44 known deaths and 170,000 evacuations. In recent years, a surge in wildfire cases lead to the research of better methods of forecasting wildfire climate. In this study, a neural network was developed, and climate data was input to a neural network to determine the current and future wildfire danger. A Feed Forward Neural Network (FFNN) for classification tasks was incrementally trained to make predictions of future fire weather. FFNN outputs were used to predict effective fire prevention solutions. The system was trained and tested using open source climate data from 2018-2019, and forest images from google.com. The FFNN uses temperature (°C), relative humidity (%), total precipitation (mm), wind speed/direction (mph), mean sea level pressure (kPa) and shortwave radiation (sfc). The classification network performs at an accuracy of 0.963, representative of high classification strength and dependability on Neural Network predictions to discern between safe, warning, and fire conditions. The cost is less than 5% of the losses of California, making the system cost efficient.

1 Introduction

The first few months of 2019 alone have seen damages caused by 40,000 wildfires, destroying a total of 3.2 million acres of forests. The total recorded cost of these wildfires was over 18 million per state. They have caused 44 known deaths and displaced 170,000 people (Facts Statistics: Wildfires, 2019). Wildland forest fires are currently among one of the leading causes of carbon dioxide (CO_2) emissions into the Earth's atmosphere. In the month of June, wildfires in the Arctic emitted as much CO_2 as the entire country of Sweden did for the first six months of 2019 (Newburger, 2019). Many wildfires are only found hours, sometimes days, after they begin. This leads to a lack of control of their spread and major damage to wildlife and contribution to climate change.

In recent years, a surge in wildfire cases led to the research better methods of forecasting wildfire climate. The governments of Canada, United States, and other developed nations, have created automated systems for the detection of wildfires. Among these are NASA's Moderate Resolution Imaging Spectroradiometer (MODIS), the Raytheon Company's Visible Infrared Radiometer Suite (VIIRS) and Canada's Forest Fire Weather Index (FWI). Details regarding the strengths and weaknesses of these three systems are outlined in Table 1.

Table 1: Description and evaluation of MODIS, VARIIS, and FWI systems.

	Input	Pros	Cons
MODIS	Images of the Earth's surface at various points along the United States. Variable attribute information for each pixel based on radiative power (Thumaty, 2015).	Real time and human-verifiable data. Low maintenance cost. Improved Geolocation and orbit stability (Csiszar et al, 2014).	Image data is only sent once per day, and is often blurry and/or low resolution (De Groot, 1998). Dependent on human verification can produce low accuracy. Very large pixel spacing, 1 km, leads to low precision.
VIIRS	Radiometric measurements in the midwave infrared range (~4μm). Has an additional day-night band (DNB) of 0.5-0.9μm (Csiszar et al, 2014).	Highly sensitive, comparably more sensitive than other systems (Thumaty et al, 2015). More sensitive to smaller and cooler fires (Csiszar et al, 2014). Data from both the daytime and nighttime orbits of the satellite (Csiszar et al, 2014).	Data delivered at 750 or 375 m space resolutions (Schroeder et al, 2014). Often results in high-ambiguity pixel outputs, rather than definite fire predictions. Only looks at one form of data, with no verification (De Groot, 1998).
FWI	Data depends solely on temperature, relative humidity, precipitation, and wind speed	Analyzes multiple forms of data, and makes comparisons between them.	Outputs are often unclear, as mapping to several functions makes it difficult to compare results between indicators (Alkhatib, 2014)

	(Dimitrakopoulos et al, 2011; Wagner, 1985).	Only method that uses multiple sensors for verification.	

Notice that these systems all have limited use. Both the MODIS and VIIRS systems are based on data from one sensor, which cannot be verified, and are taken only once per day, which means the state is unknown for 23 hours. Additionally, the data is recorded over 375 -750 m spaces in the VIIRS system and the 1km space in the MODIS system. These ranges limit the precision of the systems to changes in fire danger in smaller intervals. The FWI system has ambiguous outputs, making it difficult for humans to use it to make predictions. Additionally, all of these techniques suffer from data inaccuracies caused by wireless propagation effects.

Wireless signals, such as sensor data, propagate from a transmitter to a receiver along a channel, a multiplexed medium over which the wave travels. An example is a radio channel or a wire. As the signal propagates, the signal is amplified, reflected, and distorted (Koks & Challa, 2003). In many cases, loss or distortion of data ("noise") can lead to uncertain results or misleading information. To decrease errors and promote accurate information, methods such as data fusion and probabilistic data filtering are used (Koh, 1995). In data fusion, independent data sources are combined. In probabilistic data filtering, errors related to interactions and uncertainties between signals as they propagate are reduced (Sekkas et al, 2010). Merging data from multiple sensors has positive effects on performance, reliability, and intelligent combination of data concerning a system's state (Sasiadek & Hartana, 2000).

In the past, data fusion has been attempted using Bayesian, Kalman, and Dempster-Shafer methods. Bayesian systems assign true-false values to parameters that describe the system, but are often inefficeint. Kalman systems use a similar idea but do so with respect to past information (Sasiadek & Hartana, 2000). While Bayesian systems determine environment from two possible values of each parameter, Kalman systems assign a new parameter, based on the prior state vector of the system, to describe the current state vector, and predict future state vector. The Dempster-Shafer model does both, but also provides a third possibility for each parameter, an 'unknown' value focused on describing uncertainty in its prediction state and its future state (Koks & Challa, 2003). Another possible, although uncommonly used, method is Fuzzy Logic. Fuzzy Logic implements a 'membership function', taking sensor data as inputs and outputting a value that generalizes into a number that lies between one and zero, relating to a certain plausibility that defines how likely an element is to exist in a certain system state (Yadaiah et al, 2006). These are not useful because they all depend on extra information that may not necessarily be available in weather data, or analyze data in a way that is inapplicable to weather data.

Given the usefulness of data fusion, the use of a Neural Network with data fusion features is a good candidate for wildfire applications. Feed Forward Neural Networks (FFNN) are commonly used in machine learning because they perform better than Convolutional Neural Networks (CNNs) and Artificial Neural Networks (ANNs) in numerical classification tasks (Paik & Katsaggelos, 1992). While CNNs (Frizzi et al, 2016; Khaw et al, 1995) and ANN (Koh, 1995; Sontag, 1988; Yadaiah et al, 2016) are typically used in machine learning, FFNNs are used in this research because of data fusion features (Obayya & Abou-Chadi, 2008).

The purpose of this study is to use Feed Forward Neural Network machine learning and test its predictive ability for wildfire detection.

2 Methods

2.1 Data Sources

In the past, many FWIs have used temperature, relative humidity, total precipitation and wind speed/direction, as five inputs, to calculate fire probability (Adou et al, 2015; Van Wagner & Pickett, 1985). Two additional inputs were also included in this study, mean sea level pressure (MSLP) and shortwave radiation, for a total of seven inputs. MSLP is a good indicator of flammable content as lower MSLP is correlated to a lower ignition temperature (Zhang et al, 2018). Shortwave Radiation correlates to a higher energy content, and a higher probability of fire.

Hourly sensor data was obtained from mateoblue.com, and data was tagged in 3 classes (Safe, Warning and Fire) using information available at Wildfire Today (wildfiretoday.com), from March 2018 - July 2019.

Thirty-one thousand vector inputs of data were normalized according to a z-score normalization, so that the range of the data was between 0 and 1 (Ristic et al, 2004). As described in section 2.4, each tag had 1000 sets of data (Appendix A).

2.2 Feed Forward Neural Network Structure

Using the data discussed, a Feed Forward Neural Network (FFNN) performed all data fusion. Initially neural network patterns, such as the number of layers, the size of the layers and the weights/biases were initialized using Matlab's neural net toolbox (nntool).

For multi-dimensional classification tasks, the "patternNet" type for Matlab neural networks works better than other types (Koh, 1995; Li et al, 2019; Khaw et al, 1995; Frizzi et al, 2016; Yadaiah et al, 2006; Sontag, 1988).

A Feed Forward Neural Network (FFNN) was used for this classification task, showing in Figure 1. This model mainly uses sigmoid neurons, and the universal approximation theorem to achieve better performance and lower error. In simple terms, the universal approximation function states that any prediction accuracy is possible, if the network is large enough and has enough neurons in each hidden layer (Park & Sandberg, 1991). It does not, however, give any insight as to how large that network needs to be or how many neurons are required.

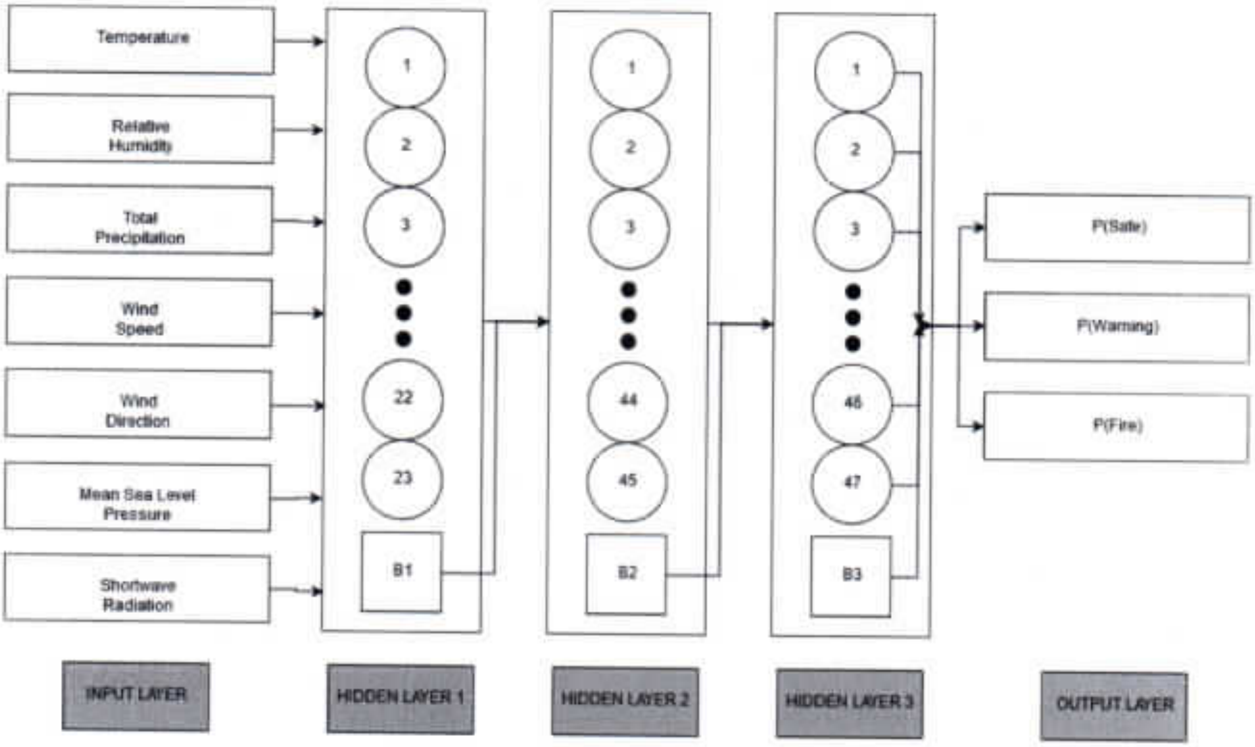


Figure 1: Feed Forward Neural Network used in this study.

Default parameters for Matlab's neural network toolbox ("nntool") were replaced by customized parameters. The performance function was changed to a cross entropy test and the hidden layer size was changed from 2 to 3 for better performance. Cross entropy is a logistic regression method commonly used in machine learning to minimize some cost function. The cost function is dependent on the target data and the output data of the network.

In order to make the outputs more probabilistic, this neural network was trained using the softmax standardization function. The standard exponential function is applied to each output, y_i, of the input vector, y, from the training matrix (Liu et al, 2016).

$$\sigma(y)_i = \frac{e^{y_i}}{\sum_{n=1}^{N} e^{y_n}}$$

$$\text{for } i = 1, \dots, N \text{ and } z = (z_1, \dots, z_N) \in \mathbb{R}^N \quad \textbf{(1)}$$

A Matlab script (Appendix B) was used to train the network at 3 different layer sizes and biases until the error between the predicted output and the known condition was minimized (Khaw et al, 1995). A shallow network was chosen because it would prevent overfitting (Yadaiah et al, 2006). Maximum possible layer sizes of fifty were chosen so that the presence of dead cells, i.e. cells with 0 weight after N iterations, would be minimized.

Of the 31,000 arrays, the network only needed ~3000 data points for training, following the 1,000 images per class rule for deep learning (Sontag, 1988). One thousand of the points were selected at random for each tag and used as the final dataset. From this data, 70% were used for training, 15% were used for validation and 15% were used for testing. The network was trained on a computer equipped with an Intel Xeon CPU E5-1620 v4 @3.50 GHz.

Stochastic gradient descent was used with mini-batches of 50. The weights and biases of the neurons and layers were initialized randomly. The minimum gradient requirement and minimum performance were 1E-6.

Decision on layer size was based on minimized confusion, optimizing performance and analysis of the Receiver Operating Characteristic (ROC) curve. Final layer sizes, L_φ, were $L_1 = 23, L_2 = 45, L_3 = 44, L_o = 3$, where L_o represents the output layer.

Outputs of the neural network were used as inputs to logic that follows in the code to determine which of the solutions to use (Table 2) to deal with the warning or existing fire.

2.3 Solutions Design

After network outputs were received, a solutions code ran in order to decide which approach to undertake. Network Output Probabilities (NOPs) were used to determine which range the output data existed in.

Since the NOPs for this task were probabilities of the three statuses, the true status was determined by comparing all three of the probabilities. Coefficients of the decision inequalities were determined by deciding where mean square error (MSE) performance error was minimized.

$$S(y) = \begin{cases} S_1, & P(1) > 5\big(P(2) \cup P(3)\big) \\ S_2, & P(1) \cup P(2) > 1.5P(3) \\ S_3, & P(2) \cup P(3) > 1.5P(1) \\ S_4, & P(3) > 2\big(P(1) \cup P(2)\big) \\ S_5, & \quad P(1) \approx P(2) \approx P(3) \end{cases} \quad \textbf{(2)}$$

The statuses of the system each have their own procedure in the solutions code. FWI calculations are based on the Canadian Fire Weather Index.

Table 2: Solutions' statuses, inequalities, and conditions.

Status	Conditions	Decision
S1	P(Safe) > 5*[P(Warning) and P(Fire)]	Data indicates safe, no action. There is no expectation that there will be a fire, nor any potential threat.
S2	P(Safe) and P(Warning) > 1.5*P(Fire)	Low risk of fire. The threat level is calculated and the estimated time before fire information is sent to user interface. If it is determined that there is a high hazard level, then Status 3 will be used.
S3	P(Warning) and P(Fire) > 1.5*P(Safe)	High risk of fire which would require further detection, perhaps via mounted cameras or drones. FWI is calculated and sent to user interface
S4	P(Fire) > 2*[P(Safe) and P(Warning)]	Definite fire class detected. FWI is calculated and the local fire department will be informed of the fire threat.
S5	P(Safe) ~ P(Warning) ~ P(Fire)	No action will be taken and the software will wait for the next set of data. Report concerning the uncertainty is sent to user interface.

3 Results & Discussion

Over the course of training and evaluation, a three-layer, multi-neuron model was chosen for this task. A FFNN (Feed Forward Neural Network) was utilized, and various parameters had to be determined to optimize performance for wildfire detection. Hidden layer size was changed from one to three because multiple layers allow the input data to be transformed to match the output layer more accurately (Paik & Katsaggelos, 1992). The first layer makes general observations about the dataset and the second layer refines those observations (Zhang et al, 2016). Inclusion of a third layer allows for more refinement. More than 3 layers led to slow processing, so the number of layers were set to 3. Outputs of the convolutional layers, or feature maps, allow the network to determine which dataset parameters affect the status of the forest the most.

The final (output) layer takes the observations and outputs of the refinement calculations in the previous layers and fits it to probabilities between 0 and 1 for each target. For example, if a dataset is of class 1 (safe), then the output of the network due to the final layer could be [.97 .02 .01], representing that there is a 97% probability that the data shows safety, with 2% and 1% being the probability that the data falls in the second (warning) and third (fire) classes, respectively.

However, this network size was chosen for two reasons. Firstly, the small layer size and layer depth means that estimated values will be output quickly, and efficiently. For neural networks with many layers, the time to calculate the output would overstep the time until the new measurement is received. This may force calculations to be done one or more arrays of data late, producing a lag in the system. Secondly, the layer size that was chosen had a high enough accuracy.

After the classification network training phase, several performance metrics were used to evaluate the network's functions. Data used for these performance metrics were datasets explicitly not part of the training dataset. One of the more common performance metrics used in network testing is a Mean Squared Error (MSE) test. Along with the MSE, the Confusion Matrix, Receiver Operating Characteristic (ROC) and the Area Under the Curve (AUC) of the ROC were assessed. The Confusion Matrix is a visual representation of the ability of the classifier to distinguish between certain classes. Higher values along the diagonal of the matrix show a high classification accuracy (Table 4).

ROC and AUC are other measures of accuracy, where a ROC which plots most of the data points in the top left corner is one that has low error. The AUC represents the percentage of the testing data that was classified correctly (Figure 2).

In addition to MSE performance and using the Confusion Matrix; Accuracy(A), Precision (P), Recall (R) and an F-score (F) were calculated. The standard equations of Accuracy, Precision and Recall are in Equation 3.

$$\begin{gathered}\text{Accuracy} = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \\ Precision_{class} = \frac{True\ Positives}{True\ Positives + False\ Positives} \\ Recall_{Class} = \frac{True\ Positives}{True\ Positives + False\ Negatives}\end{gathered} \quad \textbf{(3)}$$

Accuracy measures how often the predicted classification is correct. Accuracy shows how well the classifier can classify a data vector as being a part of one class, and how well it can classify something as not being a part of that same class. Precision measures the ability of the network to not falsely classify an output. Recall, or Sensitivity, is a measure of the completeness of the classifier. It measures how effectively the network will classify each one of the classes. A lower Recall means that the network is more likely to incorrectly assign a data vector a certain class.

From Precision and Recall, an F-score can be determined, using Equation 4. The F-score can be interpreted as a weighted average of the Precision and Recall.

$$F_{Class} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad \textbf{(4)}$$

3.1 Classification Network

The Performance of the network decreased from 0.052 inaccuracy (measured using Matlab's Neural Network MSE function) in the predicted status to 0.042 after adaptive learning. Adaptive learning is a technique that reduces the MSE by adjusting layer weights according to a fluid adjustment system, rather than a backpropagation system (Howard, 2017). Lower MSE represents more accurate outputs with smaller discrepancies with the true values. On average, the network misjudged the input data by 0.042. Because the output classes for this network were a matrix of probabilistic outputs ranging from 0 to 1, this MSE represents a 4% error in the output probabilities.

The FFNN was optimized in performance when compared to past studies, as the >93.0% performance, recall, and F-score suggest better application of the network. The accuracy of the network, performed for a testing set of 300 data vectors, was 0.963, with 289 of the data vectors being classified correctly. The Precision of the network classes, in table 3 for Fire, Warning, and Fire classes, respectively, were 95.2%, 94.1%, and 96.9%. The Recalls were 94.6%, 93.5%, 98.1%. These precisions suggest a highly reliable network output, and ability for the network to discern between class data.

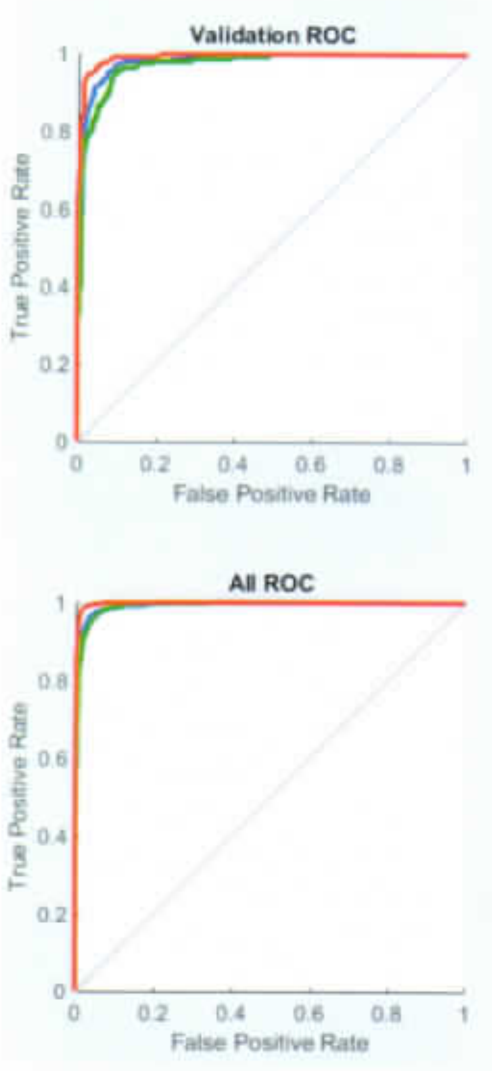
Table 3: Precision, Recall and F-scores for network classes after adaptive learning.

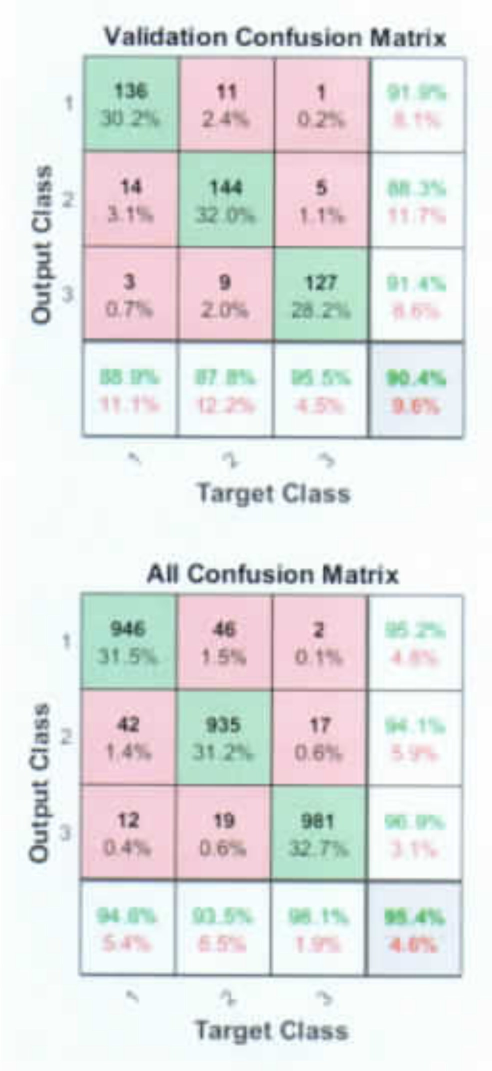
	Precision	Recall	F-score
Safe (1)	0.952	0.946	0.949
Warning (2)	0.941	0.935	0.938
Fire (3)	0.969	0.981	0.975

Of the 112 neurons, only 3 (2.67%) were weightless in the final network parameters, meaning that these neurons did not affect the network task. This percentage of weightless neurons suggests a well designed network architecture.

The ROC (Receiver Operating Characteristic) curve of the network, shown in Figure 2, had a fire AUC of 96.78% (blue), a warning AUC of 92.65% (green) and a safe AUC of 98.44% (red). The ROC and AUC of the classification network also suggest that the network has a higher, or more accurate ability to classify the sensor data. As can be seen in the ROC, the lines for Safe (Red), Warning (Blue), and Fire (Green) are very close to the top left corner. In a statistical sense, this suggest that there is a true positive rate very close to 100% and that there it is highly likely that the network will perform well when applied.

Table 4: Confusion matrix for validation and overall training steps.





Validation Confusion Matrix

Output Class \ Target Class	1	2	3	
1	136 30.2%	11 2.4%	1 0.2%	91.9% 8.1%
2	14 3.1%	144 32.0%	5 1.1%	88.3% 11.7%
3	3 0.7%	9 2.0%	127 28.2%	91.4% 8.6%
	88.9% 11.1%	87.8% 12.2%	95.5% 4.5%	90.4% 9.6%

All Confusion Matrix

Output Class \ Target Class	1	2	3	
1	946 31.5%	46 1.5%	2 0.1%	95.2% 4.8%
2	42 1.4%	935 31.2%	17 0.6%	94.1% 5.9%
3	12 0.4%	19 0.6%	981 32.7%	96.9% 3.1%
	94.6% 5.4%	93.5% 6.5%	98.1% 1.9%	95.4% 4.6%

Figure 2: ROC curves for Validation and All stages of Classification network.

When compared to the validation phase of the testing, where the smaller dataset was applied to the network to reduce its error, the ROC curves across all classes were much closer to the top left than in the overall network. In this case, this can be concluded to represent an above average performance expectancy for the network. In some cases, such as when the network is fully applied, there may be errors due to differing input arguments, or ranges of input arguments that are completely estimated. For example, if the dataset has no values for temperature from 50 to 100 degrees Fahrenheit, then there is a greater uncertainty in that range as its outputs are purely estimation based.

The Confusion matrix, in Table 4, has the Precision, Recall and Accuracy values from Table 3, but also report which mistaken classes were assigned to which reported classes. For example for the validation step, the network mistook a class 3 (fire) for safe only once out of 450 validations, which is 0.2% of the time. Notice that the primary diagonal, the green boxes of the confusion matrix, shows when the FFNN predicted the actual conditions, which is 90.4% of the time. In the testing dataset, after adaptive learning was applied, this became 95.4%.

When assessing the confusion matrix, the Precision and Recall are represented by the green percentages on the outer right and outer bottom rows. Because these values have been assessed earlier, we are more interested in the red values, the cases in which the network classification was incorrect.

Safe, Warning and Fire were incorrectly classed 11.1%, 12.2% and 4.5%, respectively, for the validation phase and 5.4%, 6.5% and 1.9% for the testing phase (bottom rows of matrices in Table 4). This suggests that the inaccuracy rate is lower at the applied neural network than when it was developing. The inaccuracy rate of the network decreased from 9.6% to 4.6% from the validation to the final network. This shows a higher applicability and a greater dependability of the network outputs.

Spread Regression was also studied to determine the relative importance of multiple fires in the same geographic region, and results yielded R^2= .09923, so application of spread regression as an indicator of the level of threat can also be used. Details on the spread regression are in Appendix C.

3.2 Forest Design and Expected Cost

In terms of application and design, sensor modules and data receivers must be well placed in flammable locations so that accurate data can be obtained. For this reason, designing a universal placement of sensors is important for applications.

Initially, the system is designed to match the sensor criteria in Figure 3. Sensors are meant to collect data that will be delivered along a line of local towers to a Central Base Unit (CBU) that will analyze data and make solutions decisions.

The data collecting mechanism is be seen in Figure 5, where a central four-direction receiver is placed equidistant to four local towers, each 17 miles away. These local towers collect data from sensor clusters up to 12 miles away. Each of these sensor clusters has a set of sensors that collect data for the FFNN.

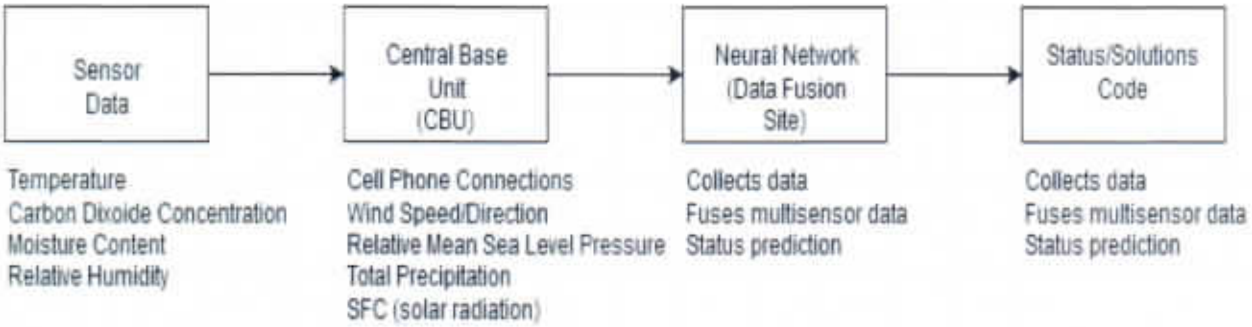


Figure 3: Flow of program and changes at each step.

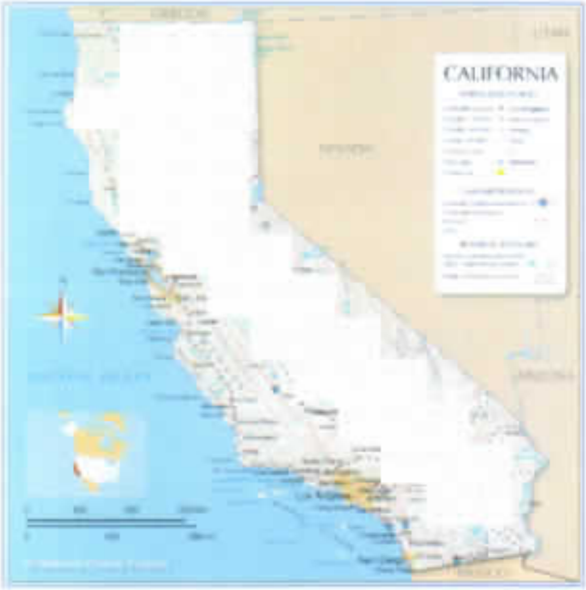


Figure 4: Sensor criteria and placement design on California. (Sensor clusters drawn to scale)

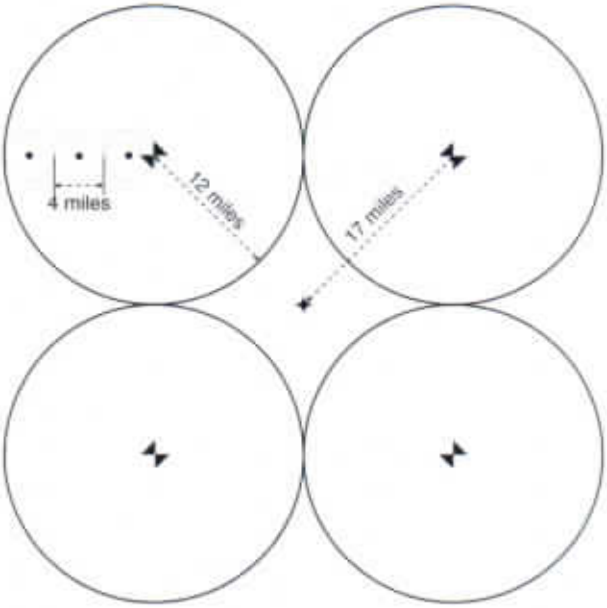


Figure 5: One cluster shown on random forest.

Forest design is done so that the least sensors would be used to cover an area. A grid design would be used to cover a 24-mile by 24-mile area, as depicted in Figure 4. Blind spots in the center and sides of the setup are negligible as they can be easily reduced by increasing the gain of sensor towers. Figure 4 describes a setup that covers an area larger than that of California's forests and relies on only 37 sensor systems.

After a set of data is collected by the CBU, it is processed and returned, through a satellite connection, to a hub that runs neural network code and determines which decision to take (Figure 3).

The cost of covered wildfire insurance losses in California alone was $9 billion in 2017 (George, 2018). However, the true economic cost of smaller wildfires, years earlier, was considered to be $500 billion

(Groden, 2015). In comparison, the total cost to implement preventive maintenance is less than the cost in damages so it makes more sense to take the proactive approach rather than the reactive one.

The 33 million acres of forest in California can be covered by 90 CBUs. This means 360 Local Towers, with 36 sensor clusters each, for a total of 13000 sensor clusters. If the cost of one sensor cluster was $10,000, then the total cost to cover all of California's forests would be $130 million dollars. This represents only 2.6% of the cost of wildfire suppression to the state currently, compared to losses that cost 20% of the total Gross Domestic Product of the state (Groden, 2015). However, a total cost of $10,000 is much higher than expected, because most sensors are $100-200 dollars and last up to 3 years when updating every 5 minutes. As technology advances, the relative cost can be expected to decrease.

4 Conclusion and Future Work

According to performance measurements for the neural network and regression function, the system has dependability, efficiency and accuracy. At a 96.3% accuracy, the ability to discern between safe, fire and warning is strong. Firstly, because the network can almost always discern between safe and fire, the system is already strong in determining whether a fire is currently occurring. Ahead of this, regression tasks and the high positive rate for the warning class show an ability to predict future fires very well. At a cost that is less than five percent of the losses of California, the state that experiences the most wildfires, the system is cost efficient.

However, no field testing was done on the system, so it's true behavior is unknown. Rather, it is known that for historical data, the system was able to predict the current state of the forest with great ease and speed. The regression is also untested, but it's extremely high CVR-squared value suggests a strong ability to determine the danger of fires that the neural network detects.

Future work includes developing an automated system to analyze images taken at the locations in which the neural network detects fires. This measure allows for greater accuracy and quicker solution. Since the MODIS and VARIIS systems have performed relatively well using only image inputs, inclusion of such a mechanism would greatly improve the performance of the system. Additionally, overlap areas where the gain increase in the towers cannot reach will become less uncertain zones if drone analysis is done. Additionally, research on more accurate and low-cost sensors can be done to continue to decrease the cost of application of the system. Developing a toolkit and User Interface for the system would also allow for human assessment of the situation, only increasing the accuracy of the system in taking measures against predicted or occurring fires.

5 References

Adou, J. K., Brou, A. D. V., & Porterie, B. (2015). Modeling wildland fire propagation using a semi-physical network model. *Case Studies in Fire Safety*, *4*, 11-18.

Alkhatib, A. A. (2014). A review on forest fire detection techniques. *International Journal of Distributed Sensor Networks*, *10*(3), 597368.

Csiszar, I., Schroeder, W., Giglio, L., Ellicott, E., Vadrevu, K. P., Justice, C. O., & Wind, B. (2014). Active fires from the Suomi NPP Visible Infrared Imaging Radiometer Suite: Product status and first evaluation results. *Journal of Geophysical Research: Atmospheres*, *119*(2), 803-816.

Culverhouse, P. F., Williams, R., Reguera, B., Herry, V., & González-Gil, S. (2003). Do experts make mistakes? A comparison of human and machine indentification of dinoflagellates. *Marine Ecology Progress Series*, *247*, 17-25.

De Groot, W. J. (1998, April). Interpreting the Canadian forest fire weather index (FWI) system. In *Proc. of the Fourth Central Region Fire Weather Committee Scientific and Technical Seminar*.

Dimitrakopoulos, A. P., Bemmerzouk, A. M., & Mitsopoulos, I. D. (2011). Evaluation of the Canadian fire weather index system in an eastern Mediterranean environment. *Meteorological Applications*, *18*(1), 83-93.

Facts Statistics: Wildfires. (2019, September). Retrieved from https://www.iii.org/fact-statistic/facts-statistics-wildfires.

Frizzi, S., Kaabi, R., Bouchouicha, M., Ginoux, J. M., Moreau, E., & Fnaiech, F. (2016, October). Convolutional neural network for video fire and smoke detection. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society* (pp. 877-882). IEEE.

George, K. (2018, December 18). The Cost Of CA Wildfire Damage In 2018 Is Massive - And It Could Keep Climbing. Retrieved from https://www.bustle.com/p/the-cost-of-california-wildfire-damage-in-2018-is-astronomical-it-could-keep-climbing-15519504.

Groden, C. (2015, September 15). What's the True Cost of the American West's Wildfires? Retrieved from https://fortune.com/2015/09/15/cost-california-wildfires/.

Howard, S. M. (2017). *Deep Learning for Sensor Fusion* (Doctoral dissertation, Case Western Reserve University).

Khaw, J. F., Lim, B. S., & Lim, L. E. (1995). Optimal design of neural networks using the Taguchi method. *Neurocomputing*, *7*(3), 225-245.

Koh, L. P. L. (1995). *A neural network approach to multisensor data fusion for vessel traffic services* (Doctoral dissertation, Monterey, California. Naval Postgraduate School).

Koks, D., & Challa, S. (2003). *An introduction to Bayesian and Dempster-Shafer data fusion* (No. DSTO-TR-1436). Defence Science And Technology Organisation Salisbury (Australia) Systems Sciences Lab.

Li, M., Hu, Y., Zhao, N., & Qian, Q. (2019). One-Stage Multi-Sensor Data Fusion Convolutional Neural Network for 3D Object Detection. *Sensors*, *19*(6), 1434.

Liu, W., Wen, Y., Yu, Z., & Yang, M. (2016, June). Large-margin softmax loss for convolutional neural networks. In *ICML* (Vol. 2, No. 3, p. 7).

Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., & Baik, S. W. (2018). Convolutional neural networks based fire detection in surveillance videos. *IEEE Access*, *6*, 18174-18183.

Newburger, E. (2019, August 17). Massive Arctic wildfires emitted more CO2 in June than Sweden does in an entire year. Retrieved from https://www.cnbc.com/2019/08/17/arctic-wildfires-emit-more-co2-in-june-than-sweden-does-in-an-entire-year.html.

Obayya, M., & Abou-Chadi, F. (2008, November). Data fusion for heart diseases classification using multi-layer feed forward neural network. In *2008 International Conference on Computer Engineering & Systems* (pp. 67-70). IEEE.

Paik, J. K., & Katsaggelos, A. K. (1992). Image restoration using a modified Hopfield network. *IEEE Transactions on image processing*, *1*(1), 49-63.

Park, J., & Sandberg, I. W. (1991). Universal approximation using radial-basis-function networks. *Neural Computation*, *3*(2), 246-257.

Ristic, B., Arulampalam, S., & Gordon, N. (2004). Beyond the Kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, *19*(7), 37-38.

Sasiadek, J. Z., & Hartana, P. (2000, July). Sensor data fusion using Kalman filter. In *Proceedings of the Third International Conference on Information Fusion* (Vol. 2, pp. WED5-19). IEEE.

Schroeder, W., Oliva, P., Giglio, L., & Csiszar, I. A. (2014). The New VIIRS 375 m active fire detection data product: Algorithm description and initial assessment. *Remote Sensing of Environment*, *143*, 85-96.

Sekkas, O., Hadjiefthymiades, S., & Zervas, E. (2010, November). A multi-level data fusion approach for early fire detection. In *2010 International Conference on Intelligent Networking and Collaborative Systems* (pp. 479-483). IEEE.

Sontag, E. D. (1988). Some remarks on the backpropagation algorithm for neural net learning. *Rutgers Center for Systems and Control Technical Report*, 88-07.

Thumaty, K. C., Rodda, S. R., Singhal, J., Gopalakrishnan, R., Jha, C. S., Parsi, G. D., & Dadhwal, V. K. (2015). Spatio-temporal characterization of agriculture residue burning in Punjab and Haryana, India, using MODIS and Suomi NPP VIIRS data. *Curr. Sci*, *109*, 1850-5.

Van Wagner, C. E., & Pickett, T. L. (1985). *Equations and FORTRAN program for the Canadian forest fire weather index system* (Vol. 33).

Vitolo, C., Di Giuseppe, F., Krzeminski, B., & San-Miguel-Ayanz, J. (2019). A 1980–2018 global fire danger re-analysis dataset for the Canadian Fire Weather Indices. *Scientific Data*, *6*, 190032.

Yadaiah, N., Singh, L., Bapi, R. S., Rao, V. S., Deekshatulu, B. L., & Negi, A. (2006, July). Multisensor data fusion using neural networks. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings* (pp. 875-881). IEEE.

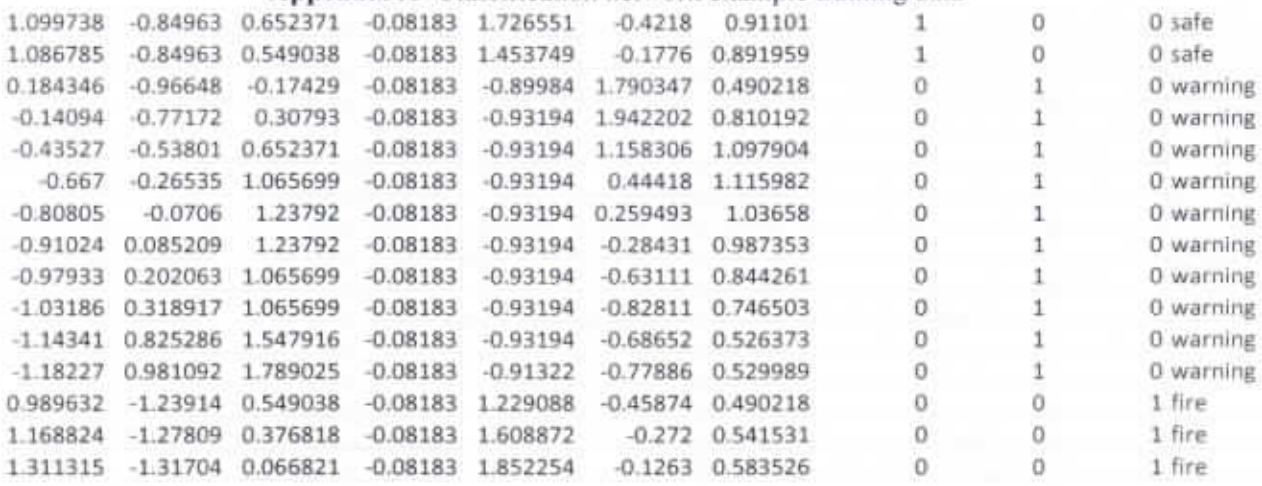
Zhang, Q. X., Lin, G. H., Zhang, Y. M., Xu, G., & Wang, J. J. (2018). Wildland forest fire smoke detection based on faster R-CNN using synthetic smoke images. *Procedia Engineering*, *211*, 441-446.

Zhang, Q., Xu, J., Xu, L., & Guo, H. (2016, January). Deep convolutional neural networks for forest fire detection. In *2016 International Forum on Management, Education and Information Technology Application*. Atlantis Press.

6 Appendix

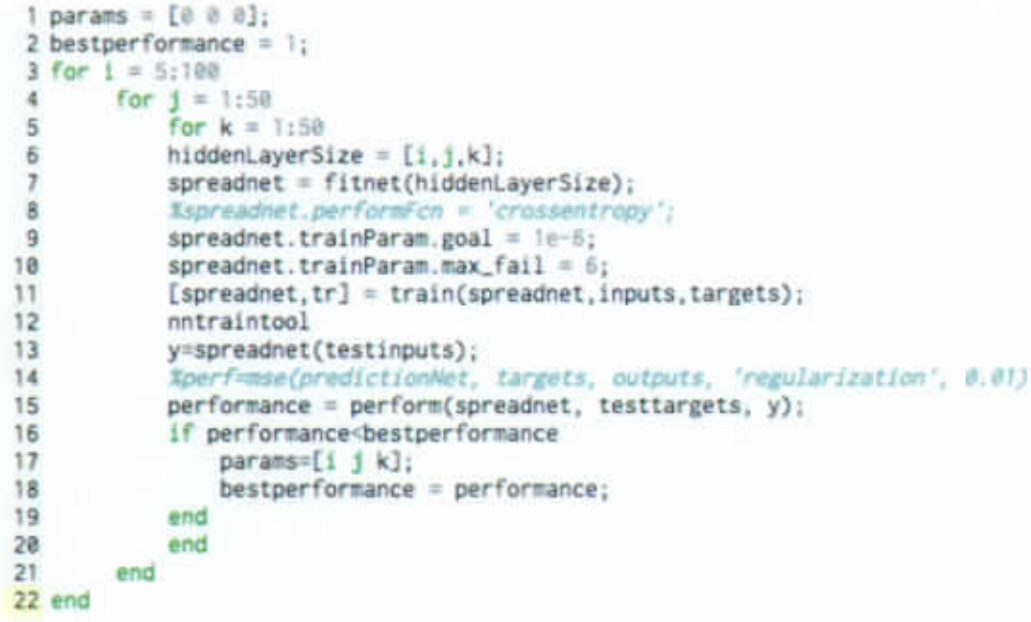
Appendices are in the order that they appear in this paper.

Appendix A -Classification network example training data



1.099738	-0.84963	0.652371	-0.08183	1.726551	-0.4218	0.91101	1	0	0	safe
1.086785	-0.84963	0.549038	-0.08183	1.453749	-0.1776	0.891959	1	0	0	safe
0.184346	-0.96648	-0.17429	-0.08183	-0.89984	1.790347	0.490218	0	1	0	warning
-0.14094	-0.77172	0.30793	-0.08183	-0.93194	1.942202	0.810192	0	1	0	warning
-0.43527	-0.53801	0.652371	-0.08183	-0.93194	1.158306	1.097904	0	1	0	warning
-0.667	-0.26535	1.065699	-0.08183	-0.93194	0.44418	1.115982	0	1	0	warning
-0.80805	-0.0706	1.23792	-0.08183	-0.93194	0.259493	1.03658	0	1	0	warning
-0.91024	0.085209	1.23792	-0.08183	-0.93194	-0.28431	0.987353	0	1	0	warning
-0.97933	0.202063	1.065699	-0.08183	-0.93194	-0.63111	0.844261	0	1	0	warning
-1.03186	0.318917	1.065699	-0.08183	-0.93194	-0.82811	0.746503	0	1	0	warning
-1.14341	0.825286	1.547916	-0.08183	-0.93194	-0.68652	0.526373	0	1	0	warning
-1.18227	0.981092	1.789025	-0.08183	-0.91322	-0.77886	0.529989	0	1	0	warning
0.989632	-1.23914	0.549038	-0.08183	1.229088	-0.45874	0.490218	0	0	1	fire
1.168824	-1.27809	0.376818	-0.08183	1.608872	-0.272	0.541531	0	0	1	fire
1.311315	-1.31704	0.066821	-0.08183	1.852254	-0.1263	0.583526	0	0	1	fire

Appendix B: Network Training Code



```
params = [0 0 0];
bestperformance = 1;
for i = 5:100
     for j = 1:50
         for k = 1:50
         hiddenLayerSize = [i,j,k];
         spreadnet = fitnet(hiddenLayerSize);
         %spreadnet.performFcn = 'crossentropy';
         spreadnet.trainParam.goal = 1e-6;
         spreadnet.trainParam.max_fail = 6;
         [spreadnet,tr] = train(spreadnet,inputs,targets);
         nntraintool
         y=spreadnet(testinputs);
         %perf=mse(predictionNet, targets, outputs, 'regularization', 0.01)
         performance = perform(spreadnet, testtargets, y);
         if performance<bestperformance
             params=[i j k];
             bestperformance = performance;
         end
         end
     end
end
```

Appendix C – Spread Regression

Spread regression was studied so that differences between fires occurring at the same time could be assessed, and the system could chose to solve one before another.FWI variable and spread data was acquired from a past study. (Vitolo et al, 2019) A dataset of 517 matrices were used for regression (Appendix B).

These calculations provide insight as to the spread of a fire once it has started, as well as the level of danger of the particular fire. Insight from these values allow the local fire department to produce a threat assessment. Meanings of the value ranges of the FWI are dependent on current and previous statuses of the forest and surrounding areas. FWI variables were used in a regression to determine a fit for the expected spread of the fire.

The regression function created the function r(x) represented in the table below. Because the function itself is too large, it has been summarized in the table.

The regression function, a 6th order polynomial function with 12012 terms. It returned an r-squared value of .9923. the CVR-squared value was -3.4191e+08. An R-squared value of .9923 represents low MSE in the predicted value and the true value of the data. A CVR-squared shows the ability of a regression to understand how small changes in related variables affects the regression's estimations. Additionally, testing data fitting returned the following scatter plot (Figure 5) Yhat represents the outputs of the regression task, and y represents the correct output.

Table 3: Spread regression function characteristics

Property	Value
Coefficient Matrix	1716 x 7 Double
Order	6
R-Squared	0.9923
CVR-squared	-3.419e+08
X-norm	1/1

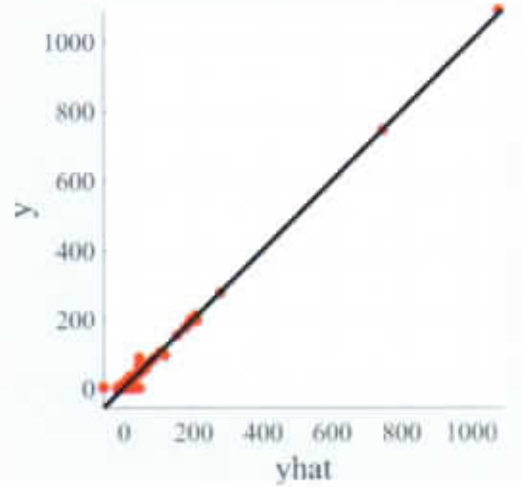
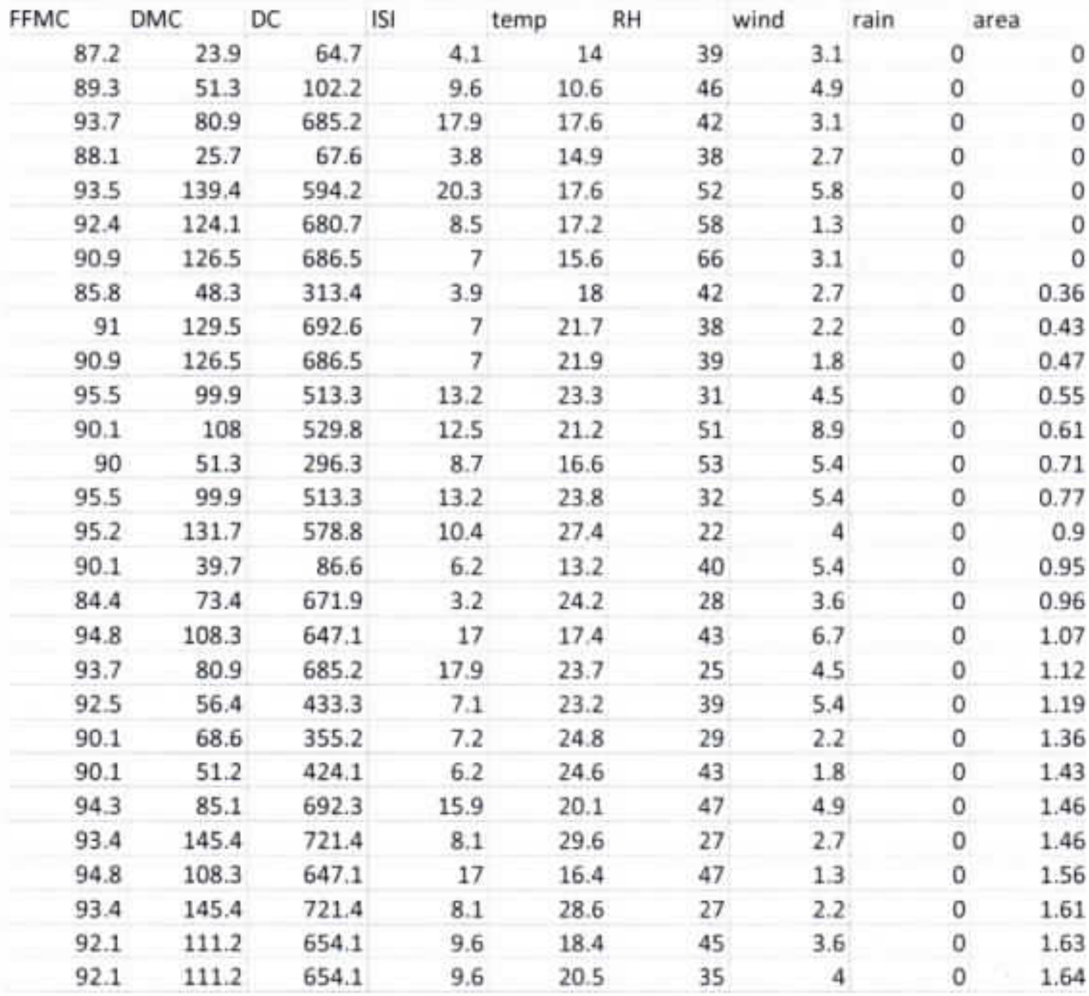


Figure 5: Regression outputs versus true values.

The spread regression network returned an r-squared value of 0.9923. Statistically, this represents a spread estimation that is highly dependable. This shows a very low error value. In the application of the system, these spread indices would be very true and very predictable, allowing the Regression function to produce good results.

The CVR-Squared value, representative of errors, suggests a higher accuracy and dependability of the regression's outputs. Since it has a very large magnitude, it is able to make connections between trends in the input data vectors of the dataset matrix very well when comparing them in clusters, rather than all at once.

Sample data vectors for training:



FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
87.2	23.9	64.7	4.1	14	39	3.1	0	0
89.3	51.3	102.2	9.6	10.6	46	4.9	0	0
93.7	80.9	685.2	17.9	17.6	42	3.1	0	0
88.1	25.7	67.6	3.8	14.9	38	2.7	0	0
93.5	139.4	594.2	20.3	17.6	52	5.8	0	0
92.4	124.1	680.7	8.5	17.2	58	1.3	0	0
90.9	126.5	686.5	7	15.6	66	3.1	0	0
85.8	48.3	313.4	3.9	18	42	2.7	0	0.36
91	129.5	692.6	7	21.7	38	2.2	0	0.43
90.9	126.5	686.5	7	21.9	39	1.8	0	0.47
95.5	99.9	513.3	13.2	23.3	31	4.5	0	0.55
90.1	108	529.8	12.5	21.2	51	8.9	0	0.61
90	51.3	296.3	8.7	16.6	53	5.4	0	0.71
95.5	99.9	513.3	13.2	23.8	32	5.4	0	0.77
95.2	131.7	578.8	10.4	27.4	22	4	0	0.9
90.1	39.7	86.6	6.2	13.2	40	5.4	0	0.95
84.4	73.4	671.9	3.2	24.2	28	3.6	0	0.96
94.8	108.3	647.1	17	17.4	43	6.7	0	1.07
93.7	80.9	685.2	17.9	23.7	25	4.5	0	1.12
92.5	56.4	433.3	7.1	23.2	39	5.4	0	1.19
90.1	68.6	355.2	7.2	24.8	29	2.2	0	1.36
90.1	51.2	424.1	6.2	24.6	43	1.8	0	1.43
94.3	85.1	692.3	15.9	20.1	47	4.9	0	1.46
93.4	145.4	721.4	8.1	29.6	27	2.7	0	1.46
94.8	108.3	647.1	17	16.4	47	1.3	0	1.56
93.4	145.4	721.4	8.1	28.6	27	2.2	0	1.61
92.1	111.2	654.1	9.6	18.4	45	3.6	0	1.63
92.1	111.2	654.1	9.6	20.5	35	4	0	1.64