Analysis of Forest Fire Data using Neural Network Rule Extraction with Human Understandable Rules

Osama Elsarrar
Department of Mathematics
West Virginia University
Morgantown, WV 26506, USA.
oelsarra@mix.wvu.edu

Marjorie Darrah
Department of Mathematics
West Virginia University
Morgantown, WV 26506, USA.
Marjorie.Darrah@mail.wvu.edu

Richard Devine
Department of Mathematics
West Virginia University
Morgantown, WV 26506, USA.
Rdevine1@mix.wvu.edu

***Abstract*—Forest fires spread fast, uncontrollably, and may leads to massive destruction. This makes the prevention of them a safety critical issue. Neural networks are a sub-area of machine learning that can be used to analyze the complex behavior of natural systems and help to predict forest fires. To make the knowledge learned by a neural network more accessible, rules can be extracted from the neural network to demystify the system behavior and directly relate inputs to outputs. In this paper, we present a Dynamic Cell Structure (DCS) neural network used for forest fire data prediction, determining which environmental factors lead to fires. We apply an intuitive rule extraction algorithm to extract understandable rules for this prediction. The results are verified through direct comparison with the raw data.**

Keywords—forest fire, neural network, rule extraction

I. Introduction

With the ever-increasing amount of data becoming available, smart data analysis is becoming pervasive in every aspect of life to solve a disparate set of problems. Machine learning seeks to reduce this range of disparate problems to a set of fairly narrow examples. The science ma- chine learning is then used to solve these examples and guarantee their solutions [1]. One example of machine learning that can be used to provide analysis for a wide range of problems is the neural network. However, some refer a neural network as a black-box method that can be difficult to understand and trust. It is also sometimes challenging to know exactly how the inputs are related to the outputs of a neural network, and whether the selected inputs have any significant relationship to the outputs [2]. There are methods, such a rule extraction, that paired with neural networks make the knowledge the neural network has learned by being trained on the data a little easier to understand and can assist with the connection between input and output more understandable.

One significant threat to the environment and human life, where analysis would be beneficial is in the area of forest fire prediction. In the past, a large effort was made to collect data and build automatic detection tools that could assist Fire Management Systems (FMS). With respect to forest fires, there are several potential methods that can be used. By utilizing meteorological approaches, satellites, and infrared/smoke scanners, the data can better predict when and where a fire could occur. Temperature, wind, relative humidity, etc. are factors that come into play when analyzing the meteorological aspect of it. Using such analyses methods help strengthen fire management techniques [3].

Several researchers have applied various method of analysis to the area of forest fire prediction. Clar, Drossel and Schwable [4] applied the idea of self-organization to the analyses forest fire data. They introduced the "forest fire of self-organized criticality" model, which refers to the tendency of certain large dissipative systems to drive themselves into a critical state independent of the initial conditions and without fine tuning of the parameters. Grishin and Filkov [5] developed a deterministic-probabilistic expert system for prediction of forest fires. Their model included the drying of forest combustibles and determined the probability of the emergence of a forest fire within the j^{th} time range of the forest-fire period (dynamic model) and fire caused by meteorological conditions.

Eskandari [6] used fuzzy sets integrated with analytic hierarchy process (AHP) in a decision-making algorithm to model the fire risk in the study area. He used four major criteria (topographic, biologic, climatic, and human factors) and 17 subcriteria in his model. The fuzzy AHP method was used to express the relative importance and priority of the major criteria and sub-criteria in forest fire risk in the study area.

Principal Component Analysis (PCA) and Self-Organizing Map (SOM) techniques have been applied to visualize and classify fire risk distribution in forest regions based on a hot spot dataset [7]. Both methods are a suitable method for extraction the high dimensional data onto a low dimensional representation. The SOM map gave an excellent classification and visualization of fire risk in forest regions via the node clusters and useful method for analysis of large size datasets. The PCA explained most the cumulative variance of data, but had difficulty with revealing a representative data pattern when the technique was applied to available large-scale data sets.

Paulo and Anible [8] used several data mining techniques for predicting size of forest fires. testing a variety of techniques, including Support Vector Machines (SVM) and random forests, and four distinct feature selection setups they achieved a predictive accuracy of 46% given a tolerance of 1ha and 61% given a tolerance of 2ha. It is worth noting that this accuracy is achieved using four independent variables.

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00308

Safi and Bouroumi [9] used backpropagation learning algorithm for predicting forest fires data. The neural network that they used is a multilayer perceptron whose number and size of hidden layers can be heuristically determined for each application using its available data examples. They improve the error rate (ER) from 25% to 9%. They fixed the Input layer to 12 neurons and the output to one neuron. Also, they used C++ to code the algorithm. I use the same data but different method.

In this work, we use a dynamic cell structure (DCS) neural network with an associated rule extraction method to analyze various meteorological and environmental input parameters. The goal is to determine from a set of given parameters, what conditions will likely result in a forest fire. The DCS does the analyses, but the rule extraction techniques are used to produce rules that can be easily understood and verified by experts. The combination of these methods produces more useful and implementable results.

The rest of the paper is organized as follows. In Section 2, we present background material on the DCS neural network and rule extraction method. Section 3 discusses the application of our technique to the forest fire data. Section 4 provides a comparison of previous forest fire analyses with our method. Section 5 give conclusions.

II. DCS Neural Network and Rule Extraction

This section discusses the Dynamic Cell Structure (DCS) neural network, the idea of rule extraction in general, and the specific rule extraction techniques developed for the DCS neural network.

A. Dynamic Cell Structure NN

One type of neural network is self-organizing map. The specific self- organizing map that we are working with is called the Dynamic Cell Structure (DCS) neural network [10, 11, 12, 13]. This type of neural network is designed as a topology representing network whose role is to learn the topology of an input space. The DCS neural network partitions the input space into Voronoi regions (Fig. 1). The neurons within the neural network represent the reference vector (centroid) for each of the Voronoi regions. The connections between the neurons, c_{ij} , are then part of the Delaunay triangulation connecting neighboring Voronoi regions through their reference vectors.

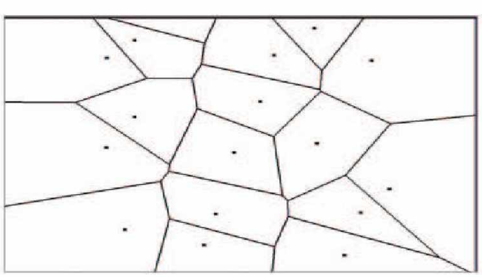

Fig. 1. Voronoi Diagram

Given an input to the DCS, v, the best matching unit (BMU) is the neuron whose weight, w, is closest to v. Along with the BMU, the neighbors of the BMU are found through the Delaunay triangulation. During adaptation, adjustments are made to the BMU and neurons within the BMU neighborhood based on the input.

The DCS algorithm consists of two learning rules, Hebbian and Kohonen (See below). These two learning rules allow the DCS neural network to change its structure to adapt to inputs. The ability to adjust neuron positions and add new neurons into the network gives the DCS neural network the potential to evolve into many different configurations.

$$c_{ab} = \left\{ \begin{array}{ll} 1 & a \in [BMU, SEC] \wedge b \in [BMU, SEC] \\ \alpha \cdot c_{ab} & \alpha \cdot c_{ab} > 0 \\ 0 & \alpha \cdot c_{ab} < 0 \\ 0 & a = b \end{array} \right\}$$

$$\Delta w_{BMU_i} = \varepsilon_{BMU}\left(v_i - w_{BMU_i}\right)$$

$$\Delta w_i = \varepsilon_{NBR}(v_i - w_i)$$

B. Rule Extraction

A negative seen when using artificial neural networking is the fact that the" knowledge" is coded as weights or activation values. This results in very few tools capable of validating the neural network process. Rule Extraction is a technique that can be used to make neural networks more understandable by assisting in revealing the internal knowledge of a trained neural network in an attempt to explain the behavior of a given neural network (or the system that it represents) by converting the network into a set of rules. Subsequently, the rules may be used instead of the neural network, since they are closer to human understanding.

The more accurate your rule extraction, the better it matches your neural network. The predictions of a network can be explained through the rules extracted from it, making a neural network less of a black box of unexplained answers and more of an understandable process [14]. By using rule extraction, the degree of matching between network responses and rule classification allows the developer and user to understand the neural network inner workings and be confident in what it has learned [15].

C. Types of Rule Extraction

Rule Extraction is a technique that can be used with several different types of classification techniques, such as decision trees, support vector machines, and neural networks. For now, the focus will be on the algorithmic methods that have been developed using the three types of rule extraction: pedagogical, decompositional, and eclectic. Each type of rule extraction focuses on different aspects of the neural network.

The pedagogical, or "black box," approach creates the rules by paying close attention to the input-output relationships, attempting to mirror the way the neural networks understand the relationship between the input-output signals as close as possible. The pedagogical approach to algorithms is typically the fastest approach because it does not take the time to scrutinize or analyze the internal weights of the network. However, because of this, this approach is also less likely to accurately obtain all of the rules that help describe the network's behavior [16]. The main advantage of using the pedagogical approach lies in the fact that is applied to most neural networks, whereas the decomposition approach can be more limited [17].

The decompositional, or "white box," approach can be more difficult than the pedagogical; however, the extra effort it takes helps improve the accuracy of the rules extracted. The de-compositional approach takes a look at the internal weights and connections that make up the network in order to more accurately extract rules [16]. The advantage of this approach is that the analyzing of the internal weights and makeup help create an accurate set of rules for the entire neural network [17].

The eclectic, or "mixed box," approach combines the ideas of the pedagogical and decompositional methods. Generally, this can take longer than the pedagogical approach because of the decompositional aspects it uses, but like the decomposition approach, the results are likely to be much more accurate than the pedagogical [16].

There are several types of rules that can be formulated from the rule extraction process. The rules can take on the form of an IF..THEN...ELSE statement, or an M-of-N statement, or If "a variable is in range" THEN "statement"[18, 19, 20].

D. DCS Rule Extraction Algorithms

The original DCS Rule Extraction algorithm was developed to generate human-readable rules that could be examined and understood by a person [17]. The second rule extraction algorithm was developed to completely capture the internal structure of the network and agree with the network 100 percent of the time [17]. This algorithm generates deterministic rules from a trained DCS that can be used in a two-step process to help refine the rules generated by the original algorithm. Although these rules are not easily understood by a human, they can be implemented and function like a fixed neural network. Both algorithms were previously applied to real-world data [18]. In this paper we will focus on the human-understandable rule extraction algorithm.

The human-understandable algorithm developed for extracting rules from the DCS was a modification of the LREX algorithm that was used to extract rules from a radial basis function neural network. Before performing the rule extraction, the DCS was put into operation for some time (learning on inputs or training), the weights on the connection were then used as input to the rule extraction algorithm. During operation, the BMU (centroid of a region) corresponding to each data point presented is recorded and then these are used as inputs to the algorithm. The data that has been presented to the neural network during operation (or training) is divided into regions based on the BMUs that have been recorded. Then for each BMU, x_{lower} is the smallest value of the independent variable and x_{upper} is the largest value of that independent variable that has that same BMU. These two numbers form bounds for the intervals in the antecedent of the rule (i.e. variable $\geq x_{lower}$ AND $\leq x_{upper}$). An interval is determined for each of the independent variables and the statements are connected by ANDs to form the full antecedent. The algorithm for extracting human-readable rules from the DCS is presented below.

Human Understandable Rule Extraction Algorithm for DCS:

Input:

Weights from a trained DCS (centers of Voronoi region)

Best matching unit for each input

Output:

One rule for each cell of the DCS

Procedure:

Apply input stimulus to DCS from training data

Record BMU for each input

Collect all inputs with common BMU to form cell

For each weight (w_i)

For each independent variable

x_{lower}= min{x | x has BMU= w_i}

x_{upper}= max{x | x has BMU= w_i}

Build rule by:

Independent variable in $[x_{lower}, x_{upper}]$

Join antecedent statements with

AND

Dependent variable = category

OR

Dependent variable in $[y_{lower}, y_{upper}]$

Join conclusion statements with

AND

Write Rule

Fig. 2 shows a two-dimensional depiction of how the rules fit with the Voronoi structure of the DCS. The human-understandable rules do not fully capture the shape of the region, but they approximate the region and encompass all data that is in the region. The downside with this approximation is that rules can sometimes overlap each other or sometimes overlap into another region. When the data is in a higher dimension, the overlap is less likely.

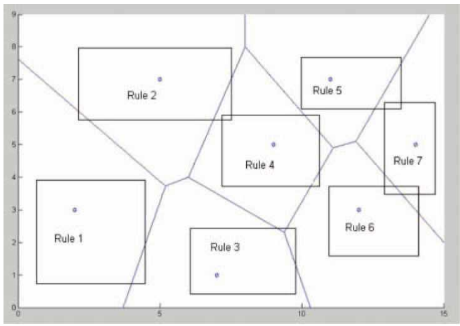

Fig. 2. Voronoi Diagram

III. TEST RESULT

A. Benchmark Testing with Iris Data

The Dynamic Cell Structure (DCS) neural network was first trained on the Iris Data. The rule extraction algorithm, written in MATLAB, was employed. The Iris data set is available from the UCI machine learning database and has four independent variables (petal width, petal length, sepal width, sepal length) and one dependent variable (type of Iris). This data set is widely used to test different algorithms. The set is interesting because it is not linearly separable. After training the DCS on the Iris data, rules were extracted by applying the algorithms to the weights and connection matrix. Below is an example of the types of rules extracted from the DCS neural network for the Iris data set.

If (SL $\geq$ 5.6 AND $\leq$ 7.9) AND (SW $\geq$ 2.2 AND $\leq$ 3.8) AND (PL $\geq$ 4.8 AND $\leq$ 6.9) AND (PW $\geq$ 1.4 AND $\leq$ 2.5)

Then Virginica

B. Forest Fire Data Set

The Forest Fire data set is available from the UCI machine learning [3]. It is composed of 517 instances and 13 attributes of data from the Montesinho Park in the Northeast region of Portugal. The aim is to use the data to predict the likelihood of a forest fire given the conditions outlined in by the parameters.

The 13 attributes included in the Forest Fire data set:

X - x-axis spatial coordinate within the Montesinho Park map:1 to 9
Y - y-axis spatial coordinate within the Montesinho Park map: 2 to 9
month - month of the year: 'Jan' to 'Dec'
day - day of the week: 'mon' to 'sun'
FFMC - FFMC index from the FWI system:18.7 to 96.20
DMC - DMC index from the FWI system: 1.1 to 291.3
DC - DC index from the FWI system: 7.9 to 860.6
ISI - ISI index from the FWI system: 0.0 to 56.10
temp - temperature in Celsius degrees: 2.2 to 33.30
RH - relative humidity in %: 15.0 to 100
wind - wind speed in km/h: 0.40 to 9.40
rain - outside rain in mm/m2: 0.0 to 6.4
area - the burned area of the forest (in ha): 0.00 to 1090.84 (for our purposes coded as 0 no fire or 1 fire occurred)

C. Analyzing Forest Fire Data

The DCS software allows for the configuration of the neural network. One of the parameters that can be chosen is the number of cells (or Voronoi regions) that will be developed during training. There is the ability to allow the neural network to grow without bound, but the result in this situation would be an overfit the neural network to the training data and provide poor generalization to future data. The best configuration is the least number of cells with the best accuracy. This allows for more general rules that can then be used more successfully with data that is not the training data.

The table below shows how the accuracy of the neural network's predictive abilities for the Forest Fire data changed with the number of cells allowed to grow in the DCS. The number of cells is treated as an independent variable and modified to create a DCS neural network with the best ability to predict forest fire occurrence. The neural network is trained on a random 75% of the data set and human-understandable rules were extracted. Then the remaining 25% of the data set was used as test data to check. The training and testing is run multiple times with different partitions of the data set each time; then an average is computed. The accuracy is judged in two ways. First, the accuracy of the neural network itself at predicting that forest fire is checked (NN accuracy). Second, the test data was processed by the rules to determine how accurate the rules were in predicting forest fire would occur (Rule accuracy).

TABLE 1 DETERMINING THE NUMBER OF CELLS FOR BEST ACCURACY.

Number cells	NN accuracy	Rule accuracy
16	0.67647	0.61765
14	0.64706	0.66667
12	0.73529	0.75758
10	0.67647	0.70588
8	0.66176	0.65306
6	0.67647	0.71698
4	0.72059	0.72414
3	0.66176	0.65574

As we see here in Table 1, when the neural network was restricted to growing only 12 cells, the neural network and the rules were the most accurate. Appendix A shows the complete set of human-understandable rules that were extracted from the neural network producing the best results when the neural network was restricted to 12 cells.

After the optimal number of nodes is established, then different subsets of the variables are used to determine if a smaller number of input variables can be used to accurately determine the output. All subsets from size two to number of independent variables (12) were run using allowing the network to grow 12 nodes. This is a large number of sets, in this case 122, so this is process is automated. Below in Table 2 the best subsets are listed.

TABLE 2: DETERMINING BEST SUBSET OF VARIABLES FOR BEST ACCURACY.

Parameters	NN Accuracy	Rule Accuracy
day, rain	0.81538	0.65672
x, y, month, day, ffmc, dmc, dc, isi, temp, rain	0.61538	0.59259
x, day	0.59231	0.54808
day, wind	0.58462	0.55963
day, wind, rain	0.58462	0.53488
x, y, month, day, ffmc, dc, isi, rh, wind, rain	0.57692	0.55446
x, y, month, day, ffmc, dc, isi, temp, rh, wind	0.57692	0.56122
x, day, rain	0.56154	0.53097
x, y, month, day, ffmc, dc, temp, rh, wind, rain	0.56154	0.55238
x, y, month, day, ffmc, dc, isi, temp, rh, wind, rain	0.54615	0.57009
x, y, isi, rh	0.53077	0.55172
x, y, isi, rh, rain	0.53077	0.54839
y, month, day, ffmc, dc, isi, temp, rh, wind, rain	0.50000	0.52427

From this table, it can be seen that using the two variables day and rain the most accurate classification of the output variable (fire occurred).

IV. CONCLUSIONS

When obtaining data for events like forest fires, there are several potential methods that can be used. By utilizing meteorological approaches, satellites, and infrared/smoke scanners, the data can better predict when and where a fire could occur. Temperature, wind, relative humidity these factors come into play when analyzing the meteorological aspect of it. All of these methods help strengthen fire management techniques [1].

The Dynamic Cell Structure (DCS) neural network helps make the neural net- working process more understandable and helps understand the rules for the classification process in forest fire data. We show how this technique can be used to extract understandable forest fire classification rules that could be used to help predict the occurrence of forest fires.

V. REFERENCES

[1] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, "Graph kernels," Journal of Machine Learning Research, vol. 11, no. Apr, pp. 1201–1242, 2010.

[2] K. Sudheer, "Knowledge extraction from trained neural network river flow models," Journal of Hydrologic Engineering, vol. 10, no. 4, pp. 264–269, 2005.

[3] P. Cortez and A. Morais, A Data Mining Approach to Predict Forest Fires using Meteorological Data, in J. Neves, M. F. Santos and J. Machado Eds., New Trends in Artificial Intelligence, Proceedings of the 13th EPIA 2007, Portuguese Conference on Artificial Intelligence, (2007), December, Guimaraes, Portugal, 512 - 523, ISBN-13 978-989-95618-0-9.2007

[4] S. Clar, B. Drossel, and F. Schwabl, "Forest fires and other examples of self-organized criticality," Journal of Physics: Condensed Matter, vol. 8, no. 37, p. 6803, 1996.

[5] A. Grishin and A. Fil'Kov, "A model of prediction of forest-fire hazard," Journal of engineering physics and thermo- physics, vol. 76, no. 5, pp. 1139–1144, 2003.

[6] S. Eskandari. "A new approach for forest fire risk modeling using fuzzy AHP and GIS in Hyrcanian forests of Iran." Arabian Journal of Geosciences 10.8 (2017): 190.

[7] S. Annas, K. Suwardi, T. Kanai, and S. Koyama. "Principal Component Analysis and Self-Organizing Map for Visualizing and Classifying Fire Risk in Forest Regions," Agricultural and Information Research, vol. 16, no. 2, pp. 44-51, 2007.

[8] P. Cortez, and AJR Morais. "A data mining approach to predict forest fires using meteorological data." (2007).

[9] S. Youssef, and A. Bouroumi. "Prediction of forest fires using artificial neural networks." Applied Mathematical Sciences 7.6 (2013): 271-286.

[10] J. Charles C. "Direct adaptive aircraft control using dynamic cell structure neural networks." (1997).

[11] B. Jörg, and G. Sommer. "Dynamic cell structure learns perfectly topology preserving map." Neural computation7.4 (1995): 845-865.

[12] F. Bernd. "Growing cell structures—a self-organizing network for unsupervised and supervised learning." Neural networks 7.9 (1994): 1441-1460.

[13] T. Martinetz, Competitive Hebbian Learning Rule Forms Perfectly Topology Preserving Maps, Proc. of the ICANN 93. p.426-438, 1993.

[14] R. Setiono, H. Liu. Understanding neural networks via rule extraction. In IJCAI (Vol. 1, pp. 480-485). 1995.

[15] G. Bologna, "A study on rule extraction from neural networks applied to medical databases," in The 4th European Conference on Principles and Practice of Knowledge Discovery (PKDD2000), Lyon, France, 2000.

[16] M. G. Augasta and T. Kathirvalavakumar, "Rule extraction from neural networks? a comparative study," in Pattern Recognition, Informatics and Medical Engineering (PRIME), 2012 International Conference on, pp. 404–408, IEEE, 2012.

[17] M. A. Darrah and B. J. Taylor, "Rule extraction to understand changes in an adaptive system," Adaptive Control Approach for Software Quality Improvement, vol. 20, p. 115, 2011.

[18] M. Darrah, B. J. Taylor, and Spiro T. Skias. "Rule Extraction From Dynamic Cell Structure Neural Network Used in a Safety Critical Application." FLAIRS Conference. 2004.

[19] MW. Craven, and JW. Shavlik. "Using sampling and queries to extract rules from trained neural networks." Machine Learning Proceedings 1994. Morgan Kaufmann, 1994. 37-45.

[20] AB. Tickle M. Orlowski, and J. Diederich. "DEDEC: decision detection by rule extraction from neural networks." QUT NRC (1994).

VI. APPENDIX A

Human Understandable Rules From DCS

RULES FOR CELL1
IF (X ≥ 1 AND ≤ 9) AND
IF (Y ≥ 2 AND ≤ 6) AND
IF (Month ≥ 6 AND ≤ 8) AND
IF (rain ≥ 0 AND ≤ 0) AND
IF (RH ≥ 27 AND ≤ 82) AND
IF (temp ≥ 14.2 AND ≤ 28.3) AND
IF (wind ≥ 0.9 AND ≤ 8.9) AND
IF (day ≥ 1 AND ≤ 7) AND
IF (FFMC ≥ 90.1 AND ≤ 95.5) AND
IF (DMC ≥ 51.2 AND ≤ 180.4) AND
IF (DC ≥ 411.8 AND ≤ 542) AND
IF (ISI ≥ 6.2 AND ≤ 14.7)
THEN.1

RULES FOR CELL2
IF (X ≥ 1 AND ≤ 8) AND

IF (Y ≥ 3 AND ≤ 6) AND
IF (Month ≥ 7 AND ≤ 9) AND
IF (rain ≥ 0 AND ≤ 0) AND
IF (RH ≥ 27 AND ≤ 78) AND
IF (temp ≥ 10.3 AND ≤ 29.3) AND IF (wind ≥ 2.7 AND ≤ 8) AND
IF (day ≥ 3 AND ≤ 7) AND
IF (FFMC ≥ 87.1 AND ≤ 91.6) AND IF (DMC ≥ 253.6 AND ≤ 291.3) AND IF (DC ≥ 768.4 AND ≤ 860.6) AND IF (ISI ≥ 4 AND ≤ 10.1)
THEN.1

RULES FOR CELL3
IF (X ≥ 1 AND ≤ 9) AND
IF (Y ≥ 2 AND ≤ 6) AND
IF (Month ≥ 6 AND ≤ 7) AND
IF (rain ≥ 0 AND ≤ 0) AND
IF (RH ≥ 25 AND ≤ 90) AND
IF (temp ≥ 10.6 AND ≤ 28) AND
IF (wind ≥ 1.8 AND ≤ 9.4) AND
IF (day ≥ 1 AND ≤ 7) AND
IF (FFMC ≥ 53.4 AND ≤ 94.3) AND IF (DMC ≥ 49.5 AND ≤ 150.3) AND IF (DC ≥ 200 AND ≤ 377.2) AND IF (ISI ≥ 0.4 AND ≤ 56.1)
THEN.1

RULES FOR CELL4
IF (X≥1 AND ≤9) AND
IF (Y≥2 AND ≤9) AND
IF (Month≥2 AND ≤12) AND
IF (rain≥0 AND ≤0) AND
IF (RH≥21 AND ≤78) AND
IF (temp≥2.2 AND ≤27.2) AND IF (wind≥1.3 AND ≤8.5) AND
IF (day≥1 AND ≤7) AND
IF (FFMC≥79.5 AND ≤93.5) AND IF (DMC≥25.4 AND ≤85.3) AND IF (DC≥349.7 AND ≤395) AND IF (ISI≥1.5 AND ≤16.8)
THEN.1

RULES FOR CELL5
IF (X≥1 AND ≤8) AND
IF (Y≥2 AND ≤6) AND
IF (Month≥8 AND ≤9) AND
IF (rain≥0 AND ≤0) AND
IF (RH≥21 AND ≤59) AND
IF (temp≥16.2 AND ≤32.6) AND IF (wind≥0.4 AND ≤6.7) AND
IF (day≥1 AND ≤7) AND
IF (FFMC≥90.2 AND ≤96.2) AND IF (DMC≥99.6 AND ≤191.4) AND IF (DC≥624.1 AND ≤674.4) AND IF (ISI≥5.8 AND ≤17)
THEN.1

RULES FOR CELL6
IF (X≥1 AND ≤8) AND
IF (Y≥3 AND ≤6) AND
IF (Month≥8 AND ≤10) AND
IF (rain≥0 AND ≤0) AND
IF (RH≥24 AND ≤78) AND
IF (temp≥11.2 AND ≤27.8) AND IF (wind≥1.3 AND ≤7.6) AND
IF (day≥1 AND ≤7) AND
IF (FFMC≥63.5 AND ≤94.3) AND IF (DMC≥32.8 AND ≤91.8) AND IF (DC≥664.2 AND ≤726.9) AND
IF (ISI≥0.8 AND ≤22.6)
THEN.0

RULESFORCELL7
IF(X≥2AND≤8)AND
IF(Y≥2AND≤6)AND
IF(Month≥3AND≤11)AND
IF(rain≥0AND≤0)AND
IF(RH≥18AND≤80)AND
IF (temp≥5.5 AND ≤18.8) AND IF (wind≥0.9 AND ≤9.4) AND
IF(day≥1AND≤7)AND
IF (FFMC≥79.5 AND ≤94) AND IF (DMC≥3 AND ≤52.2) AND IF (DC≥67.6 AND ≤106.7) AND IF (ISI≥1.1 AND ≤12.5)
THEN.0

RULESFORCELL8
IF(X≥2AND≤9)AND
IF(Y≥2AND≤9)AND
IF(Month≥1AND≤4)AND
IF(rain≥0AND≤0)AND
IF(RH≥19AND≤82)AND
IF (temp≥4.6 AND ≤17.6) AND
IF(wind≥1.8AND≤9.4)AND
IF(day≥1AND≤7)AND
IF (FFMC≥75.1 AND ≤93.4) AND
IF(DMC≥3AND≤25.7)AND
IF(DC≥7.9AND≤67.6)AND
IF(ISI≥1.9AND≤12.3)
THEN.0

RULES FOR CELL9
IF (X≥1 AND ≤8) AND
IF (Y≥2 AND ≤6) AND
IF (Month≥9 AND ≤9) AND
IF (rain≥0 AND ≤0) AND
IF (RH≥24 AND ≤86) AND
IF (temp≥9.8 AND ≤25.3) AND
IF (wind≥1.8 AND ≤7.2) AND
IF (day≥1 AND ≤7) AND
IF (FFMC≥87.9 AND ≤92.8) AND
IF (DMC≥55.2 AND ≤136.9) AND
IF (DC≥721.1 AND ≤822.8) AND
IF (ISI≥3.7 AND ≤12.5)
THEN.1

RULES FOR CELL10
IF (X≥1 AND ≤9) AND
IF (Y≥2 AND ≤9) AND
IF (Month≥8 AND ≤8) AND

IF (rain≥0 AND ≤6.4) AND
IF (RH≥26 AND ≤88) AND
IF (temp≥10.4 AND ≤33.3) AND
IF (wind≥0.9 AND ≤5.8) AND
IF (day≥1 AND ≤7) AND
IF (FFMC≥90.5 AND ≤96.1) AND
IF (DMC≥181.1 AND ≤248.4) AND
IF (DC≥643 AND ≤753.8) AND
IF (ISI≥6.3 AND ≤18)
THEN.0

RULES FOR CELL11
IF (X≥1 AND ≤8) AND
IF (Y≥2 AND ≤6) AND
IF (Month≥8 AND ≤9) AND
IF (rain≥0 AND ≤0) AND
IF (RH≥15 AND ≤73) AND
IF (temp≥13.1 AND ≤30.2) AND
IF (wind≥0.9 AND ≤8.5) AND
IF (day≥1 AND ≤7) AND
IF (FFMC≥90.9 AND ≤94.5) AND
IF (DMC≥124.1 AND ≤175.1) AND IF (DC≥680.7 AND
≤752.6) AND
IF (ISI≥7 AND ≤20)
THEN.1

RULES FOR CELL12
IF (X≥1 AND ≤9) AND
IF (Y≥2 AND ≤9) AND
IF (Month≥7 AND ≤8) AND
IF (rain≥0 AND ≤0) AND
IF (RH≥22 AND ≤96) AND
IF (temp≥5.1 AND ≤33.1) AND
IF (wind≥1.3 AND ≤7.6) AND
IF (day≥1 AND ≤7) AND
IF (FFMC≥85.6 AND ≤95.2) AND
IF (DMC≥90.4 AND ≤181.3) AND
IF (DC≥560 AND ≤624.2) AND
IF (ISI≥6.3 AND ≤22.7)
THEN.1