

Application of neural networks to rate of spread estimation in shrublands

Rahul Wadhvani ^a, Duncan Sutherland ^b, Khalid A Moinuddin ^a, Jason J. Sharples ^b

^a Centre for Environmental Safety and Risk Engineering, Victoria University, Melbourne, Australia

^b School of Science, University of New South Wales, Canberra, Australia

Email: duncan.sutherland@adfa.edu.au

Abstract:

Wildfire behaviour prediction is a complex and challenging endeavour that incorporates information relating to various factors such as vegetation, local and ambient weather conditions, and topography within a particular modelling framework. Often, a single model cannot provide a complete description of the phenomenon due to significant differences in the temporal and spatial frames required to resolve the relevant physical process. Because of their low computational cost, statistical-based models of fire spread are an obvious choice for operational purposes. However, most statistical regression models of rate of spread still possess significant degrees of. Despite considerable progress in modelling fire behaviour, fires can still create unexpected scenarios for emergency services during real situations, and result in loss of containment, significant injury, or even fatalities. Recently, there has been significant interest in utilising machine learning techniques to better predict wildfire for improved management of forests and parks. The present work uses an artificial neural network to estimate the rate of fire spread in shrubland. The model is trained upon data from the published literature to establish a non-linear relation between the rate of fire spread and twelve input variables, namely: vegetation height, vegetation cover, fine dead and live fuel load, air temperature, relative humidity, wind speed at 2 m height, slope, moisture content of dead and live fuel, and ignition length. The network model-based estimates of the rate of fire spread yield results close to the measured data, with a correlation coefficient $r^2=0.87$. Furthermore, the model was able to quantify the significance of all input variables on rate of fire spread. This capability of neural network model can be helpful in determining how much a factor contribute to rate of fire spread which otherwise be overshadowed by the assumptions involved in a regression model. For this dataset, the moisture content of dead fuel, wind speed at 2 m height, and vegetation height were the most significant variables for determining the rate of fire spread in shrublands. This agrees with other statistical regression model studies in the literature. The neural network model was also able to quantify the dependency on ignition line length a variable that was deemed insignificant by the existing regression model. This preliminary study provides a benchmark for further application of neural networks and other machine learning techniques to model wildfire rate of spread. Further work could also involve testing the predictive capability of the neural network model on other independent datasets pertaining to shrubland fires and its extension to other fuel types.

Keywords: Artificial neural network (ANN), Machine learning, Fire spread, Shrubland

1. INTRODUCTION

Wildfire is a complex phenomenon; its occurrence and behaviour are the product of several interconnected factors, which include the type of ignition source, types of fuel involved and their composition, local and ambient weather conditions, and topography. The science of fire behaviour prediction is established in some parts of the world, however, even in countries with prominent investment in fire behaviour research and operational fire aptitude (e.g. situational awareness, fire spread forecasting, wildfire disaster policies, etc.), fires can still surprise fire and emergency agencies and sometimes cause fatalities. Recent Californian fires 2020 in the USA, Black Summer 2019-20 in Australia, and the Evia Island fires in Greece 2021 are prime examples of such scenarios, which perplexed the local emergency services. One of the most challenging aspects of the wildfire problem is the fact that fire propagation is driven by a variety of processes that encompass a wide range of spatiotemporal scales: the combustion process occurs at a spatial scale of a few centimetres over a period of seconds, whereas fire spread and growth occurs over spatial scales from meters to kilometres and over time frames of minutes to days. Thus, it is not possible to have a single model for all the physical processes contributing to wildfire propagation.

Sullivan (2009a, 2009b, 2009c) provided detailed information about the features of existing fire behaviour and propagation models used both in academia and operational fire prediction. Every modelling approach has its own inherent strengths and limitations. To begin with, empirical fire models (Sullivan, 2009b) are straightforward to implement and use, and neither require high technical skills nor computational budget. These fire models have proved to be fairly accurate, under certain restrictions due to their simplified nature, and can predict some aspects of fire-spread dynamics in an acceptable way. However, most of these models are derived from controlled experiments conducted under relatively benign conditions, and so can become unreliable for fires that take place under different conditions and in different landscapes, especially for local fire dynamic behaviours. Physics-based or quasi-physics based fire models (Sullivan, 2009a) are mostly based on first principles. These models are intrinsically nonlinear, multi-scale, and capture complex phenomena, however, their prediction accuracy cannot always be ensured. Additionally, these models require high technical skills to implement and need considerable computational resources to support simulation. The computational time is typically far greater than the simulated time, making such models unsuitable for tactical intervention scenarios. Cruz and Alexander (2013) have evaluated the accuracy of various empirical-based fire spread models, used mainly in Australia, and observed that the mean error in estimating fire spread varied between 20-310%. Uncertainties associated with fire spread modelling have significant implications for the utility of model predictions. For example, uncertainty in rate of rate spread predictions relating to a wildland-urban-interface (WUI) fire scenario, could lead to significant over-prediction in the amount of time available for evacuation, or significant under-estimation of the behaviour of the fire as it impacts the WUI (Ronchi *et al.*, 2017).

There is always significant uncertainty associated when forecasting natural hazards behaviour such as flooding, landslides, avalanches, and wildfires due to their inherent complexity and dominance of particular factors in each situation (Riley *et al.*, 2016). Mandel *et al.* (2005) proposed a flood forecasting technique that was based on a neural network model which shows the relative importance of different environmental variables used to predict floods. Kim *et al.* (2015) developed a time-dependent surrogate model for storm surge to use for forecasting evacuation and facility closures from impact of hurricanes using a neural network. More recently, there has been significant interest in the research community in utilising machine learning techniques for forecasting such events and estimating their behaviour as these techniques can increase computational processing capability and data availability. This interest has also extended to the wildfire research community, and, Jain *et al.* (2020) have provided a detailed review of the application of machine learning in the discipline of wildfire science and emergency preparation. They observed that most of the work is focused only on fire susceptibility mapping and management aspects – other aspects of the fire modelling problem, such as understanding fire spread and fire growth, have received less attention from the research community. Vakalis *et al.* (2004) developed a two-fold reasoning GIS (Geographic Information System) simulator, which consists of a discrete contour propagation model for estimating fire consequences and a fuzzy-neural system for the estimation of fire spread as a function of influencing factors such as terrain characteristics, vegetation type and density and meteorological conditions. They developed their fire spread model based on area burnt and a set of influencing factors that occurred during 1983-1996 in Attica, Greece. Their work did not involve developing a fire spread model based on individual fire events. Markuzon and Kolitz (2009) tested several classifier methods such as random forests, Bayesian networks and k-nearest neighbour to estimate if a fire would become large either one or two days following its initial observation. They concluded that each of the tested methods performed similarly, with random forest based classification performing correctly by classifying large fires at a rate of over 75%, albeit with a number of false positive cases. Kozik *et al.* (2013) presented a fire spread model that used a novel artificial neural network implementation that more closely resembled a cellular automata model than a traditional neural network.

Zheng *et al.* (2017) simulated fire spread by integrating a cellular automata model with an extreme learning machine (a type of feed-forward neural network). Transition rules for the automata model were determined by the extreme learning machine, which was trained with data obtained from five historical fires in the Western parts of the United States, as well as vegetation, topographic, and meteorological data. Recently, Hodges and Lattimer (2019) trained a convolutional neural network to predict fire spread using environmental variables (topography, weather, and fuel-related variables). Outputs of this model were spatial grids corresponding to the probability that the grid burns on a map. Their method achieved a mean precision of 89% and mean sensitivity of 80% with reference 6 hourly burn maps computed using the semi-empirical based FARSITE simulator.

The present research attempts to fill the gap in previous applications of neural network techniques by developing a rate of spread model that can be used in place of current operational models. To carry out such work requires a large amount of data, verification and validation. This work discusses a preliminary study carried out using artificial neural network (ANN) on the dataset for shrub fires published by Anderson *et al.* (2015). The objective of this work is to demonstrate the capability of ANNs to model linear and non-linear systems without the need to make implicit assumptions about the functional form of models, as is required in most traditional statistical approaches.

2. NEURAL MODEL

ANNs can be grouped into two major categories: feed-forward and feedback (or recurrent) networks. In the feed-forward networks, no loops are formed by the network connections, while one or more loops may exist in the feedback networks. The most commonly used family of feed-forward networks is a layered network (commonly called the multilayer perceptron (MLP)) in which nodes called neurons are organised into layers with connections strictly in one direction from one layer to another. A feed-forward network is used in the present study. Fig.1 represents a typical architecture of a feed-forward ANN which comprises three layers i.e. input, hidden and output layers. An input layer consists of input nodes, with each node associated with each input variable. The output layer is similar, with each node corresponding to each desired output variable. Defining hidden layers is crucial, as there are no set rules nor conditions to define the number of hidden layers nor the number of nodes in each hidden layer (Stathakis, 2009). However, heuristic rules suggest that one to two hidden layers are sufficient for most engineering applications of neural networks. Higher numbers of hidden layers are usually useful only in the deep-learning methods applied to a very complex data set (Beale *et al.*, 2010). The number of nodes in the hidden layer is usually selected between the number of input and output nodes. Heaton (2008) suggested that the number of hidden layers should not exceed two and the number of nodes in a hidden layer should be the mean of the number of nodes in input and output layers.

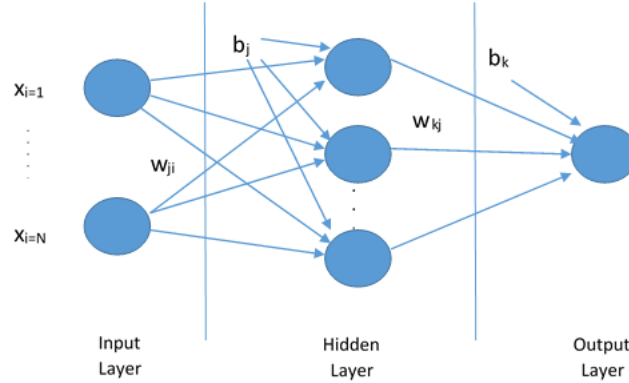


Fig. 1: Architecture of feed-forward layered network type artificial neural network

The detailed working procedure and theory of ANN are provided by Heaton (2008) and Beale *et al.* (2010). Only MLP feed-forward networks are discussed here since they are used in the present study. In MLP networks, neuron nodes in the input layer only act as buffers for distributing the input signals x_i ($i = 1, 2, \dots, N$) to neurons in the hidden layer. Each neuron node j (Fig. 1) in the first hidden layer sums up its input signals x_i after weighting them with the strengths of the respective connections w_{ji} from the input layer and computes its output y_j as a transfer function f of the sum, including some bias b_j . f can be a simple linear function or a sigmoidal function, or it can be a more sophisticated hyperbolic tangent or a radial basis function.

$$y_j = f \left[\sum_{i=1}^N w_{ji} x_i + b_j \right] \quad (1)$$

The output of neurons in the output layer is computed similarly.

The weights are determined by training the network upon a data set. The back-propagation algorithm, a gradient descent algorithm, is the most commonly adopted MLP training algorithm. It gives the change Δw_{ji} of the weight of a connection between neurons i and j as follows:

$$\Delta w_{ji} = \eta \delta_j x_i \quad (2)$$

where η is called the learning rate, and δ_j is a factor depending on whether the neuron node j is connected to an output node or hidden layer node which is defined as follows:

For the output layer node,

$$\delta_{ji} = \frac{\partial f}{\partial net_j} (y_j^{(t)} - y_j). \quad (3)$$

For hidden layer node,

$$\delta_{ji} = \frac{\partial f}{\partial net_j} \left(\sum_q w_{jq} \delta_q \right) \quad (4)$$

In Eq. (3), net_j is the total weighted sum of input signals to neuron nodes j and $y_j^{(t)}$ is the target output for neuron node j . As there are no target outputs for hidden neurons, in Eq. (4), the difference between the target and actual output of a hidden neuron nodes j is replaced by the weighted sum of the δ_q terms already obtained for neurons q connected to the output of j .

The process begins with the output layer, the δ term is computed for neurons in all layers and weight updates are determined iteratively for all connections. The weight updating process can happen after the presentation of each training pattern (pattern-based training) or after the presentation of the whole set of training patterns (batch training). A single step of training, called an epoch, is completed when all training patterns have been presented once to the MLP.

A commonly adopted method to speed up the training is to add a momentum term to Eq. (5) which effectively lets the previous weight change influence the new weight change:

$$\Delta w_{ij}(I + 1) = \eta \delta_j x_i + \mu \Delta w_{ij}(I) \quad (5)$$

Where $\Delta w_{ij}(I + 1)$ and $\Delta w_{ij}(I)$ are weight changes in epochs $(I + 1)$ and (I) , respectively, and μ is the momentum coefficient.

3. DATA AND MODEL IMPLEMENTATION

Anderson *et al.* (2015) presented a set of compiled data for a wide range of heathlands and shrubland to develop an empirical fire spread model. Their spread model was found to predict fire spread rate within acceptable limits against independent data from prescribed fires and wildfires. The present work utilises a subset of the data compiled by Anderson *et al.* which have complete information of each quantity used to develop their model. They observed that the rate of fire spread in shrubland and heathlands are dependent on twelve variables, namely, vegetation height, vegetation cover, amount of fine dead & live fuel load, air temperature, relative humidity, wind speed at 2 m height, slope, moisture content of dead and live fuel, and ignition length. MATLAB R2016b (Beale *et al.*, 2010) was used to create an ANN architecture for training, testing and validating. A MATLAB script was written to develop an MLP feed-forward neural network and calculate the performance of the network. Twelve input variables on which rate of spread depends are used as input nodes in the input layer while the rate of spread is considered as an output variable. Most researchers consider various

combinations of hidden layers and the number of nodes in the hidden layer to find a suitable architecture that best interprets their data. The input data are normalised ($\hat{x}_i$) according to Eq. 6.

$$\hat{x}_i = \frac{x_i - \mu}{\sigma}, \quad (6)$$

where μ is the mean and σ is the standard deviation of the x –values, respectively.

The input data were ordered randomly and then subdivided in the ratio of 60:20:20 for training, testing and validating. The Levenberg-Marquardt algorithm (Beale *et al.*, 2010) was used for training the network, which is useful for small or medium-sized datasets and is one of the gradient descent method available in MATLAB. The training was carried out using a MLP feed-forward network using a tan-sigmoidal transfer function described by Eq. 7.

$$\text{tansig}(n) = \frac{2}{(1 + \exp(-2n))} - 1 \quad (7)$$

4. RESULTS AND DISCUSSION

A set of network parameters were tested using different combinations of the number of hidden layers and number of nodes in the hidden layers based on Heaton's recommendations (Heaton, 2008). Table 1 shows the performance of six different MLP feed-forward neural networks tested for the Anderson *et al.* (2015) data set. There is little difference between the different neural architectures tested on the present dataset. The mean absolute error between the output and estimated data using neural networks shows slight improvement when an additional hidden layer is considered. However, the improvement is minimal. Details of each neural network parameter (weights and biases of each node for each model) will be available in the public domain at <https://github.com/rdwadhwani0220/shrubfire-neural.git>.

Table 1: Statistical performance of developed MLP models for different network structures

Model no.	No. of Hidden layers	No. of nodes in the hidden layers	r^2	Mean absolute error
1	1	6	0.88	2.826
2	1	12	0.87	2.652
3	2	6-6	0.87	3.033
4	2	6-12	0.88	2.183
5	2	12-6	0.87	1.852
6	2	12-12	0.89	2.446

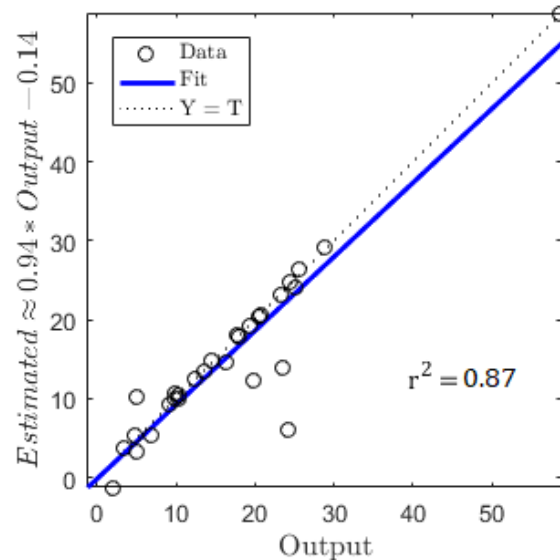


Fig. 2: Scatter plot of neural network (model no. 5) between the estimated rate of spread using neural network model and output layer variable (rate of spread data by Anderson *et al.*). $Y=T$ refers to the line where estimated is equal to the output.

Fig. 2 illustrates the performance of the neural network in estimating the rate of spread against the field data provided by Anderson *et al.* (2015) used in the output layer node of the neural network. In this plot any biases in the ANN results were corrected by applying an affine transformation to the output data. The coefficients of the transform were computed using linear regression. In most cases the ANN estimated rate of spread is similar to Anderson *et al.*'s data. However, some outliers are observed and they are suspected due to small dataset used in the present study. A bigger dataset would assist in better development of neural network based model. The weights that connect variables in a neural network are partially analogous to parameter coefficients in a standard statistical regression model and thus can be used to describe relationships between variables. Garson (1991) proposed a method to identify the relative importance of variables for specific output variables in a supervised neural network by deconstructing the model weights. The relative importance (or strength of association) of a specific variable for a specific output variable can be determined by identifying all weighted connections between the nodes of interest. That is, all weights connecting the specific input node that passes through the hidden layer to the specific output variable are identified. The weights dictate the relative influence of information that is processed in the network such that input variables that are not relevant in their correlation with an output variable are suppressed by their weights. The contrary effect is seen for weights assigned to explanatory variables that have strong, positive associations with an output variable.

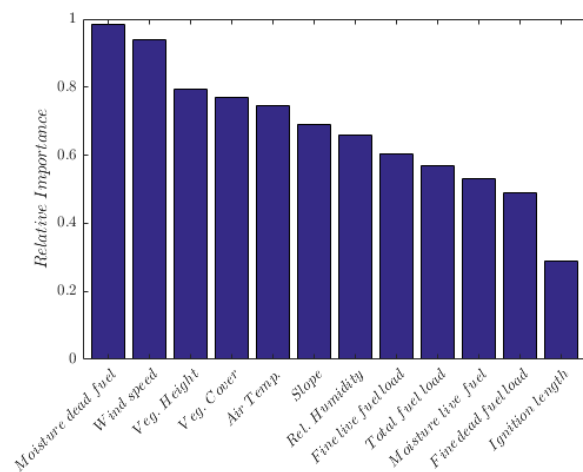


Fig. 3: Relative importance of each of the input variable on the output variable (rate of spread)

Fig. 3 shows the relative importance of each of the twelve input variables in estimating the rate of fire spread in shrubland and heathland vegetation. The weights are normalised between 0 and 1 by the maximum weight associated with the input variable. It is observed that the rate of spread is most affected by moisture content in dead fuel load and wind speed at 2 m height followed by vegetation heights and vegetation coverage. This finding is also supported by Anderson *et al.* (2015) where they found a strong correlation between dead fuel moisture and wind speed followed by vegetation height and total fuel load. However, they found that the total fuel load has more effect in their spread model compared to the vegetation cover. While our model found a more profound impact of vegetation cover in estimating the rate of fire spread. Moreover, the statistical analyses of Anderson *et al.* (2015) could not find any significant dependency of the rate of spread on the ignition length, which is contrary to other research (Cheney *et al.*, 1993). Our neural network quantifies the dependency on ignition length, which is visible by its relative importance in Fig. 3.

5. CONCLUSION

Neural network techniques have been the subject of significant research interest in past decades especially in the prediction and forecast of natural hazards. The present work shows preliminary application of neural networks in estimating the rate of fire spread in shrubland and heathland. A MATLAB based neural network model is developed based on data available in the literature. The model, which was developed from a limited data set shows good accuracy in estimating the rate of fire spread. Importantly, this model is developed with little input on the part of the modeller, whereas in statistical regression models, the modeller enforces functional dependencies for different input variables. Furthermore, the model can quantify the dependencies between the output and input variables thus assisting existing regression statistical modelling approaches which may not be able to capture such dependencies. Larger datasets pertaining to fires in shrublands can expand the capabilities and efficacy for prediction situations. Future works will involve collecting more field data and utilisation of different machine learning techniques that would assist in prediction of rate of fire spread with operational fire models.

6. REFERENCES

- Anderson, W. R., *et al.* (2015). A generic, empirical-based model for predicting rate of fire spread in shrublands. *International Journal of Wildland Fire*, 24(4), 443-460.
- Beale, M. H., *et al.* (2010). Neural network toolbox. *User's Guide, MathWorks*, 2, 77-81.
- Cheney, N., *et al.* (1993). The influence of fuel, weather and fire shape variables on fire-spread in grasslands. *International Journal of Wildland Fire*, 3(1), 31-44.
- Cruz, M. G., & Alexander, M. E. (2013). Uncertainty associated with model predictions of surface and crown fire rates of spread. *Environmental Modelling & Software*, 47, 16-28.
- Garson, D. G. (1991). Interpreting neural network connection weights.
- Heaton, J. (2008). *Introduction to neural networks with Java*: Heaton Research, Inc.
- Hodges, J. L., & Lattimer, B. Y. (2019). Wildland fire spread modeling using convolutional neural networks. *Fire Technology*, 55(6), 2115-2142.
- Jain, P., *et al.* (2020). A review of machine learning applications in wildfire science and management. *Environmental Reviews*, 28(4), 478-505.
- Kim, S.-W., *et al.* (2015). A time-dependent surrogate model for storm surge prediction based on an artificial neural network using high-fidelity synthetic hurricane modeling. *Natural Hazards*, 76(1), 565-585.
- Kozik, V., *et al.* (2013). Adaptive prediction of forest fire behavior on the basis of recurrent neural networks. *Optoelectronics, Instrumentation and Data Processing*, 49(3), 250-259.
- Mandal, S., *et al.* (2005). *A neural network based prediction model for flood in a disaster management system with sensor networks*. Paper presented at the Proceedings of 2005 International Conference on Intelligent Sensing and Information Processing, 2005.
- Markuzon, N., & Kolitz, S. (2009). *Data driven approach to estimating fire danger from satellite images and weather information*. Paper presented at the 2009 IEEE Applied Imagery Pattern Recognition Workshop (AIPR 2009).
- Riley, K., *et al.* (2016). Uncertainty in Natural Hazards, Modeling and Decision Support: An Introduction to This Volume. *Natural Hazard Uncertainty Assessment: Modeling and Decision Support*, 223, 1.
- Ronchi, E., *et al.* (2017). *e-Sanctuary: Open Multi-Physics Framework for Modelling Wildfire Urban Evacuation* (FPRF-2017-22). Quincy, MA, USA, .
- Stathakis, D. (2009). How many hidden layers and nodes? *International Journal of Remote Sensing*, 30(8), 2133-2147.
- Sullivan, A. L. (2009a). Wildland surface fire spread modelling, 1990–2007. 1: Physical and quasi-physical models. *International Journal of Wildland Fire*, 18(4), 349-368. doi:10.1071/WF06143
- Sullivan, A. L. (2009b). Wildland surface fire spread modelling, 1990–2007. 2: Empirical and quasi-empirical models. *International Journal of Wildland Fire*, 18(4), 369-386. doi:10.1071/WF06142
- Sullivan, A. L. (2009c). Wildland surface fire spread modelling, 1990–2007. 3: Simulation and mathematical analogue models. *International Journal of Wildland Fire*, 18(4), 387-403. doi:10.1071/WF06144
- Vakalis, D., *et al.* (2004). A GIS based operational system for wildland fire crisis management I. Mathematical modelling and simulation. *Applied Mathematical Modelling*, 28(4), 389-410.
- Zheng, Z., *et al.* (2017). Forest fire spread simulating model using cellular automaton with extreme learning machine. *Ecological Modelling*, 348, 33-43.