A Rule-Based Framework for Predicting Rainfall Using Decision Trees and Fuzzy Petri Nets

Meher Taj S*, Sudha M

Department of Mathematics, AMET Deemed to be University, Chennai

*Corresponding author E-mail: mehertaj.s@ametuniv.ac.in

doi: https://doi.org/10.21467/proceedings.173.6

ABSTRACT

In a crucial component of meteorology is weather prediction, and precise forecasting has a big impact on everyday life, agriculture, and disaster relief. Based on past meteorological data, this study investigates the possibility of data mining approaches for forecasting rainfall and other weather conditions. We apply the C4.5 decision tree algorithm to a dataset of 2,500 weather observations in order to categorize meteorological conditions and find trends that affect the likelihood of rainfall. We use fuzzy Petri nets to handle the inherent uncertainty in meteorological data, enabling a more adaptable, sophisticated understanding of probabilistic weather patterns. Our combined strategy provides a well-balanced framework that makes use of the C4.5 algorithm's strong classification capabilities while improving the model's dependability and interpretability via fuzzy logic. According to preliminary findings, this hybrid strategy offers a viable way to increase the precision of weather forecasts. By demonstrating how data mining and fuzzy algorithms can efficiently support short-term weather forecasting, this study advances the area of data-driven meteorology.

Keywords: Fuzzy petri nets, C4.5 algorithm, evaluation, rule generation, weather prediction

1 Introduction

Many industries, from transportation and everyday decision-making to agriculture and crisis management, depend on accurate weather forecasting. The need for sophisticated forecasting techniques that can manage intricate and erratic meteorological data is increasing as climatic trends become more unpredictable. Although they work well, traditional forecasting techniques frequently have trouble capturing the significant degree of unpredictability that comes with weather patterns [1]. When predicting rainfall, this unpredictability is particularly noticeable because even slight variations in temperature, humidity, or wind speed can have a significant impact on the results. Data mining, which uses past data to find patterns and produce accurate forecasts, has become a viable strategy in this regard. A popular data mining technique for classification problems, the C4.5 decision tree algorithm is well-suited to forecasting discrete meteorological outcomes like the incidence of rainfall. It can find important rainfall predictors because of its decision tree form, which offers an understandable and straightforward perspective of the correlations between many meteorological variables. However, a strictly deterministic approach would not be enough due to the complexity of weather data, which frequently involves overlapping or ambiguous circumstances. We incorporate fuzzy Petri nets into our model to overcome this constraint, providing an additional degree of adaptability and interpretability that enables the management of data uncertainties.

Fuzzy Petri nets are perfect for scenarios where the relationships between variables are not strictly binary since they use fuzzy logic, which is an extension of conventional Petri nets. With the use of fuzzy logic's adaptive skills and the accuracy of the C4.5 algorithm, we hope to develop a model that can not only properly forecast weather conditions but also decipher patterns with a level of detail that is not possible with conventional techniques. This hybrid method is intended to improve the accuracy of weather forecasts and offer more detailed insights into weather behavior. It has been tested on a dataset of 2,500 weather observations.

Proceedings DOI: 10.21467/proceedings.173; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-970666-x-x

2 Related works

The related efforts on rule-based techniques, applications of the C4.5 algorithm, and fuzzy Petri nets for weather prediction and related topics are the main emphasis of this section. Among the many machine learning applications in meteorology covered in this survey are rule-based and decision-tree methods such as C4.5 for interpretive weather prediction [2], which are important for short- and medium-term forecasts. This study highlights the use of rule-based machine learning to derive actionable insights from weather data by focusing on generating rules from sizable observational datasets to improve forecasting's interpretability. Using frameworks such as fuzzy Petri nets to capture confusing weather patterns and decision-making, a number of studies investigate fuzzy logic models in handling the inherent ambiguity in meteorological data.

3 Methods and Materials

In this study, fuzzy Petri nets and the C4.5 decision tree algorithm are used in a hybrid approach to forecast weather conditions based on historical data. Preparing the data, building the model in the WEKA environment using the C4.5 algorithm, and including fuzzy Petri nets to account for weather pattern uncertainty are the three main steps of the methodology. Each phase is described in detail below.

3.1 Data Set

The experimental data in this paper is gained through kaggle. This dataset provides a practical introduction to machine learning for beginners, particularly for those exploring classification techniques. With 2,500 weather observations, it offers a straightforward yet valuable resource for predicting rainfall based on a range of weather conditions. Its simple structure makes it ideal for those new to classification problems and interested in analyzing weather data. This dataset has 5 input attributes and 1 decision class: they are Temperature, Humidity, Wind-speed, Cloud cover, Pressure and Rain.

3.2 Data Preprocessing

The study's weather dataset comprises 2,500 recordings documenting various meteorological characteristics, including temperature, humidity, wind speed, and precipitation frequency. To get the data ready for analysis, we carried out several preprocessing procedures before implementing machine learning algorithms

3.3 Classification with the C4.5 Algorithm in WEKA

The C4.5 decision tree technique was selected due to its interpretability and efficiency in addressing classification challenges. To optimize information gain at each node, C4.5 recursively splits the dataset according to attribute values to create a decision tree [3]. The following are crucial actions to take when implementing C4.5 in the WEKA environment.

Data Import and Configuration: The preprocessed dataset was loaded into WEKA, an open-source machine learning tool that supports C4.5 under the "J48" classifier. This tool was chosen for its user-friendly interface and wide range of machine-learning algorithms.

Model Training: To avoid overfitting and guarantee generalizability, we specify a confidence factor for pruning while configuring the J48 classifier. Eighty percent of the dataset was used to train the model, with the remaining twenty percent set aside for testing.

Evaluation measures: The model was assessed using measures like accuracy, precision, recall, and F1 score following training. The effectiveness of the C4.5 algorithm in classifying rainfall events may be examined thanks to these measures, which offer a thorough evaluation of the model's performance.

3.4 Incorporating Fuzzy Petri Nets for Uncertainty Management

While C4.5 provides a robust classification framework, weather prediction often involves inherent uncertainties, such as the variability in temperature and humidity fluctuations. To address these ambiguities, we incorporated fuzzy Petri nets (FPN) into the model. Fuzzy Petri nets allow for a degree of flexibility by incorporating fuzzy logic into the transition states, enabling the model to manage non-binary relationships between weather attributes.

Design of Fuzzy Petri Net: The FPN model was designed to reflect the relationships between key meteorological variables (e.g., temperature, humidity, wind speed) and their influence on rainfall. Each node in the FPN represents a specific condition or attribute value, with fuzzy transition rules governing the likelihood of transitioning from one state to another.

Integration with C4.5 Output: The results from the C4.5 decision tree serve as inputs to the FPN model, providing initial classifications that the FPN then refines by assessing uncertainty levels in each attribute. For instance, if the C4.5 algorithm predicts a high likelihood of rainfall but with close-to-threshold values for humidity or temperature, the FPN evaluates these inputs within a fuzzy framework to produce a more nuanced prediction.

Implementation and Tuning: The FPN model parameters were tuned to optimize the balance between prediction accuracy and interpretability [4]. Various membership functions and fuzzy rules were tested to ensure that the FPN accurately captured the relationships between the meteorological variables without over-complicating the model.

This combined approach leverages the classification power of the C4.5 algorithm while enhancing interpretability through the integration of fuzzy Petri nets. By addressing both classification and uncertainty management, the model provides a more reliable framework for predicting weather conditions. This methodology demonstrates a practical, data-driven approach for incorporating both deterministic and probabilistic reasoning, thus improving the accuracy and adaptability of weather forecasting models.

4 Results and Discussion

Import the dataset into WEKA by converting it to the ARFF (Attribute-Relation File Format) if it's not already in that format. Explore the dataset to understand its attributes, including any missing values or inconsistencies. Use WEKA's "Replace Missing Values" filter to fill any gaps in the data. Apply the "Normalize" filter in WEKA to scale continuous variables between 0 and 1, ensuring attributes contribute proportionally. Run an attribute evaluator (such as Correlation Attribute Eval) to determine which features are most relevant for predicting rainfall and filter out uninformative ones.

4.1 Classification with C4.5 Algorithm (J48 in WEKA)

In WEKA, select the J48 classifier (the implementation of C4.5). Set parameters, such as the confidence factor for pruning (default is 0.25), to prevent overfitting and improve model interpretability. Split your dataset into a training set (e.g., 80%) and a testing set (e.g., 20%). Train the J48 model on the training set by clicking "Start" after configuring the classifier. Use the test set to evaluate accuracy, precision, recall, and

F1 score, assessing how well the model classifies rainfall. Measures the proportion of correct positive predictions.

$$\text{Precision} = \frac{TP}{TP+FP}$$

High precision means fewer false alarms (false positives). Same as TPR; measures the ability of the model to capture actual positives.

$$\text{Recall} = \frac{TP}{TP+FN}$$

```
=== Summary ===

Correctly Classified Instances        2499               99.96   %
Incorrectly Classified Instances         1                0.04   %
Kappa statistic                          0.9982
Mean absolute error                      0.0004
Root mean squared error                  0.02
Relative absolute error                  0.1819 %
Root relative squared error              6.035  %
Total Number of Instances             2500

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.997    0.000    1.000      0.997    0.998      0.998    0.998     0.997     rain
                 1.000    0.003    1.000      1.000    1.000      0.998    0.998     1.000     no rain
Weighted Avg.    1.000    0.003    1.000      1.000    1.000      0.998    0.998     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
  313    1 |    a = rain
    0 2186 |    b = no rain
```

Figure 1: Generating Rules from C4.5 Model

Table 1: Confusion matrix

Actual/Predicted	**Yes**	**No**
Yes	**313**	**1**
No	**0**	**2186**
	Precision=1.0	**Recall=0.997**

In WEKA's J48 classifier setup, choose "Output rules" and "Output trees." This will show both a rule-based representation and the categorization tree. To see the produced rules when the classifier has finished running, navigate to the "Classifier Output" panel. The criteria (attributes) that result in a prediction (such as rainfall or no rainfall) are displayed in a "If-Then" manner for each rule [5]. Arrange the rules in a document by manually copying each one or, if WEKA allows exporting, saving the output. Fuzzy Petri nets will be constructed using each rule as input, with conditions standing in for transitions [6].

Proceedings DOI: 10.21467/proceedings.173
ISBN: 978-81-970666-x-x

Series: AIJR Proceedings
ISSN: 2582-3922

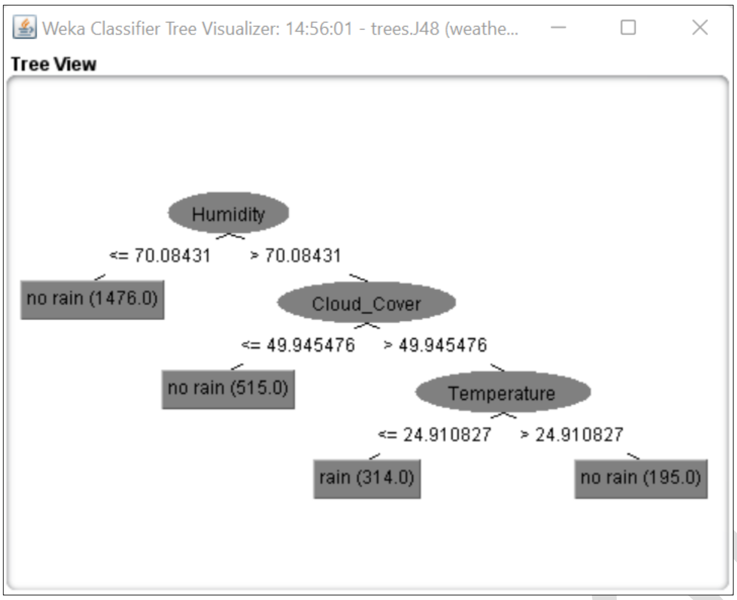

Figure 2: Decision Tree generated by J48

4.2 Fuzzy Petri Nets (FPN)

Petri Nets (PN) is a graphical and mathematical modeling tool applicable to many systems. There are promising tools for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic [7]. Fuzzy Petri nets contain two types of nodes: places and transitions, where circles represent places and rectangles represent transitions. Each place represents an antecedent or consequent and may or may not contain a token associated with a truth degree between zero and one which speaks for the amount of trust in the validity of the antecedent or consequent. Each transition representing a rule is associated with a certainty factor value between zero and one. The certainty factor represents the strength of the belief in the rule [8, 9].

4.2.1 Mapping the Rule Base to FPN

Throughout this mapping technique, all principle is represented as transitions with its relating certainty factor and each antecedent is displayed by an input place and therefore the consequents are incontestable by an output place with scrutiny truth degrees. During this displaying a transition here a suggestion is enabled to be fired if its entire input place have a truth degree resembling or over a predefined limit esteem.

Use each “If-Then” rule from the C4.5 output as a transition in the fuzzy Petri net. The conditions in each rule (e.g., temperature > 30°C and humidity < 50%) act as the input places, while the prediction outcome (e.g., Rainfall = Yes) is the output place. For each condition in the rules, define fuzzy membership functions (e.g., “High Temperature,” “Moderate Humidity”) to allow for partial matching and manage uncertainty. Create a Petri net graph where each place represents a fuzzy condition, and transitions represent the if-then relationships defined in the rules. Test the FPN by simulating it with different input conditions to ensure that it produces similar predictions as the C4.5 model while capturing the nuances of fuzzy logic.

Using the WEKA tool, Decision Tree generated by J48 produced the following rules

R1: Humidity <=70.08431 => No rain

R2: Humidity >70.08431and Cloud_ Cover <=49.945476 =>No rain

R3: Humidity >70.08431and Cloud_ Cover > 49.945476 and Temperature <=24.910827 => rain

R4: Humidity >70.08431and Cloud_ Cover > 49.945476 and Temperature > 24.910827 => No rain

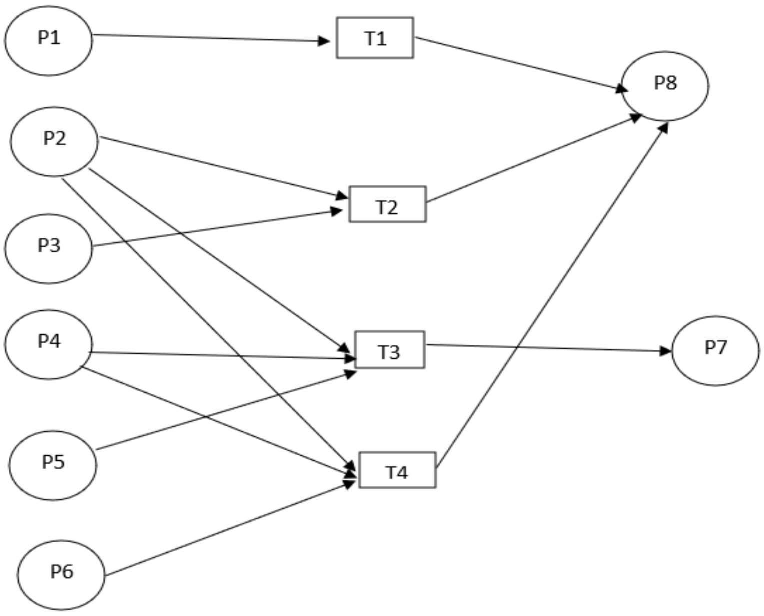

Figure 3: *Fuzzy Petri nets graph for the rules*

5 Conclusion

To increase the precision and interpretability of rainfall event classification using meteorological data, this work proposes a hybrid approach to weather prediction by integrating fuzzy Petri nets with the C4.5 decision tree algorithm. Our findings show that a strong framework for handling intricate interactions in weather forecasting is offered by this integrated method. The fuzzy Petri nets provide a nuanced representation of probabilistic transitions, improving the model's capacity to account for minute fluctuations in conditions, while the C4.5 algorithm effectively identifies meteorological conditions. By enabling flexible interpretation and adaptation to real-world uncertainties, this methodology not only increases prediction accuracy but also makes a valuable contribution to data-driven meteorology. To improve its forecasting ability, this model can be expanded in subsequent research by adding more meteorological characteristics and honing fuzzy membership functions. A promising avenue for future research in weather prediction and other fields with intricate data patterns, this study highlights the potential of combining machine learning with fuzzy systems for applications involving high levels of uncertainty.

6. Declarations

6.1 Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6.2 Acknowledgements

The authors are also thankful to AMET, deemed to be a university for their constant support and encouragement.

References

[1] N. O. Bushara, & Abraham and A. “Computational Intelligence in Weather Forecasting: A Review”, *Journal of Network and Innovative Computing*, vol. 1, no. 1, pp. 320-331, 2014, Corpus ID: 18041072.

[2] Chen L, Han B, Wang X, Zhao J, Yang W, Yang Z. “Machine Learning Methods in Weather and Climate Applications: A Survey”. *Applied Sciences.* Vol. 13, no. 21, pp 12019, 2023, https://doi.org/10.3390/app132112019.

[3] N. Hasan, N. C. Nath, and R. I. Rasel, "A support vector regression model for forecasting rainfall", *2015 2nd International Conference on Electrical Information and Communication Technology (EICT)*, pp. 554-559, 2015, doi: 10.1109/EICT.2015.7392014.

[4] Holmstrom Mark, Dylan Liu, and Christopher Vo, "Machine learning applied to weather forecasting", *Meteorol. Appl*, pp. 1-5, 2016.

[5] Nalluri Sravani, Ramasubbareddy Somula and G Kannayaram, "Weather Prediction Using Clustering Strategies in Machine Learning", *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 5, pp. 1977-1981, May. 2019, https://doi.org/10.1166/jctn.2019.7835.

[6] C.I. Christodoulou, S.C. Michaelides, M. Gabella and C. S. Pattichis, "Prediction of rainfall rate based on weather radar measurements", *IEEE*, vol. 2, pp. 1393-1396, 2004, https://doi.org/10.1109/IJCNN.2004.1380153.

[7] Murata, T, “Petri nets: Properties, analysis and applications”. Proceeding of the IEEE, l7 (4) pp. 541-580, 1989.

[8] Fei Liu, Ming Yang and Peng Shi, “Verification and Validation of Fuzzy Rules-Based Human Behavior Models”, 7th International Conference on System Simulation and Scientific Computing, 2008 pp. 813 – 819, https://doi.org/10.1109/ASC-ICSC.2008.4675474.

[9] Kouzehgar, M., M.A.Badamchizadeh and S. Khanmohammadi, “Fuzzy Petri nets for human behavior verification and validation”. *Int. J. Adv. Comput. Sci. Appl*., vol. 2, no. 12, pp. 106-114, 2011, https://dx.doi.org/10.14569/IJACSA.2011.021216.