ISSN 2319-7595

International Journal of Information Systems and Computer Sciences, Vol.7. No.1, Pages : 01-07 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJISCS/static/pdf/Issue/icsic2017sp14.pdf

Thunderstorms Prediction using Genetic Programming

Ruba Al-Jundi, Mais Yasen, Nailah Al-Madi
Department of Computer Science
Princess Sumaya University for Technology
Amman, Jordan
rub20130123@std.psut.edu.jo, mai20130045@std.psut.edu.jo, n.madi@psut.edu.jo

ABSTRACT
Thunderstorms prediction is a major challenge for efficient flight planning and air traffic management. As the inaccurate forecasting of weather poses a danger to aviation, it increases the need to build a good prediction model. Genetic Programming (GP) is one of the evolutionary computation techniques that is used for classification process. Genetic Programming has proven its efficiency especially for dynamic and nonlinear classification. This research proposes a thunderstorm prediction model that makes use of Genetic Programming and takes real data of Lake Charles Airport (LCH) as a case study. The proposed model is evaluated using different metrics such as recall, F-measure and compared with other well-known classifiers. The results show that Genetic Programming got higher recall value of predicting thunderstorms in comparison with the other classifiers.

Keywords: Evolutionary computation, Genetic Programming, Machine Learning, Weather Prediction.

1. INTRODUCTION

A major challenge for efficient flight planning and air traffic management is the accurate forecasting of weather that poses a danger to aviation [2]. The weather is a primary contributing factor in 23% of all aviation accidents [8]. Also, one important reason that causes precautionary landing, airplane crashes and delay in flights is thunderstorms.

Thunderstorms adversely affect humans and infrastructure. In the USA, lightning is the third leading cause of storm-related deaths. An estimated 24,000 deaths and 240,000 injuries worldwide are attributable to lightning [7]. Based on averages of thunderstorm events during the period 1985 to 2014, each event caused more than one billion US dollar damage [7]. A thunderstorm or convective storm is caused by a cumulonimbus cloud that produces the electric discharge known as lightning, and it typically generates heavy rainfall, gusty surface wind, and possibly hail [7].

Thunderstorms have a severe effect when it comes to aircraft, it can cause losing the lives of people. Furthermore, that effect is reflected on the economy where many airports lose money from delaying a flight or losing a flight. Thus, predicting thunderstorms in a higher accuracy is vital.

Weather stations have millions of records of the data obtained every day, which is a big data problem, the previous information that was recorded in the Metar and SYNOP (Surface Synoptic observation) data files of Lake Charles airport (KLCH) will help in the process of developing a machine learning algorithm trained and tested by this big data to increase the accuracy of forecasting.

Machine learning (ML) is a type of artificial intelligence that provides computers the ability to learn without being explicitly programmed [13]. The process of machine learning is similar to that of data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension as it is the case in data mining applications, machine learning uses that data to detect patterns and adjust program actions accordingly [13]. Supervised ML algorithms can apply what has been learned in the past to new data. One of the important data mining tasks is classification. Which aims of building a classification model based on previous data to classify new data. This classifier model can be evaluated using different measures, such as accuracy, recall, F-measure.

Evolutionary computation techniques are used for optimization problems, one of them is Genetic Programming (GP) that utilizes the same properties of natural selection found in biological evolution [9]. The mechanism of GP is to start with a collection of functions and randomly combine them into programs; then run the programs and evaluate their results, then apply natural process to produce new programs and keep this process until it gives the best results.

GP is considered a flexible algorithm since it has a variable length of its programs the only problem is that it grows in a high complexity. Since it rarely produces invalid states -if it happens they will be discarded- it uses an explicit structure to avoid any operator presence. Programs in GP are represented in tree structures which makes it easier to implement it recursively.

This paper proposes implementing GP as a classification problem to build the best classifier to predict thunderstorms. To evaluate the fitness of the produced model we used F-measure as a fitness function for GP process. F-measure was used, because the problem we have is based on imbalanced data, to avoid over-fitting problem that can exist using this accuracy measure. To best evaluate our proposed approach, we applied few well-known classifiers on the data and compared their results using different evaluation metrics. The contribution of this paper could be summarized by:

1. Using real and recent data for the first time (Lake Charles Metar and SYNOP data (LCH)).
2. Applying GP on LCH data for the first time.

ISSN 2319-7595

International Journal of Information Systems and Computer Sciences, Vol.7. No.1, Pages : 01-07 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJISCS/static/pdf/Issue/icsic2017sp14.pdf

3. Adapting GP fitness function to the thunderstorm prediction problem and using f-measure instead of accuracy.

The paper is structured as follows: Section II describes the related work in the area of weather prediction using machine learning. In Section III, genetic programming is outlined followed by the proposed approach. Section IV presents the experiments and the results of proposed approach, and Section V concludes this paper.

2. RELATED WORK

The related work shows that weather prediction is a very interesting topic to research, moreover thunderstorm prediction is rarely studied. As mentioned in [1] the authors used a time series based temperature prediction model using integrated Backpropagation (BP)/ Genetic algorithm (GA) technique, the data used is for the Ludhiana city of Punjab (India), results showed that the BP algorithm suffers from many problems, such as minima, scaling, long training time problems, they suggested solving them by using GA.

The authors in [2] propose a technique of temperature forecasting by using Feed-forward Artificial Neural Network (ANN). The data taken was from (Kaul) Haryana. The gradient descent algorithm is generally very slow because it requires small learning rates for stable learning. The momentum variation is usually faster than simple gradient descent, because it allows higher learning rates while maintaining stability, but it is still too slow for many practical applications, also data is trained by Levenberg–Marquardt algorithm. Results of this paper reflect that Levenberg BP has proven better learning rate than BP algorithms.

In experiments of A. Grover, A. Kapoor and E. Horvitz [3], the authors used deep belief network (DBN). They saw that the DBN led to an additional 1-2% error reduction. After establishing the superiority of the data-centric kernel and the DBN independently, they evaluated the prediction accuracy of the full deep hybrid model for each weather variable, aggregated over all stations in the continental US. Experiments showed that the new methodology can provide better results than National Oceanic and Atmospheric Administration (NOAA) benchmarks.

The authors of [4] demonstrated storm prediction with the help of random forest machine learning approach that examined the specific problem of combining various NWP (Numerical Weather Prediction) model, radar, satellite and derived fields for forecasting thunderstorm. For this purpose, a machine learning method that creates random forests ensembles of weak and weakly-correlated decision trees was used to rank predictor importance. The random forest approach may also be used to help identify “regimes”, the forecast logic for each regime is run, and the results were combined based on the membership values.

As showed in [5] this research did wind speed prediction through the combination of support vector regression and forecast model. With an emphasis on the support vector regression (SVR) method and its nu-SVR variant. The results showed that wind speed prediction through the combination of support vector regression and the weather research and forecast model can outperform the weather research and forecasting (WRF) in the case of the Berkeley Yacht Club data and the amount of previous day variables needed did not exceed 15 days.

The authors W. Collins, and P. Tissot in [6] developed a feed-forward multi-layer perceptron artificial neural network (ANN) to predict thunderstorm occurrence that used feed-forward multi-layer perceptron ANN topology. Two sets were developed; one set was developed after feature selection based on correlation-based feature selection (CFS). The other models were developed based on all 43 predictors. Authors proposed to compare their model performance with multi-linear regression (MLR) models, and to human forecasters National Digital Forecast Database (NDFD). Results reveal that with respect to at least one skill-based performance metric, the TANN model’s performance exceeded that of the MLR models and NDFD.

In the experiments of W. Collins and P. Tissot [7] artificial neural network (ANN) model classifiers were developed to generate $\leq$ 15 hour predictions of thunderstorms. The feed-forward, multilayer, scaled conjugate gradient learning algorithm, and the sigmoid (linear) transfer function in the hidden (output) layer were used. Three sets of ANN models were developed: two sets based on predictors chosen from feature selection (FS) techniques, and one set with all 36 predictors had three models. The best performers were a function of prediction hour domain, and FS technique. Training the models using a small fraction of the data set reduced model calibration time yet resulted in lower performance skill. Also, increasing the fraction of total positive target data in the training set did not improve generalization.

This paper proposes implementing GP to predict thunderstorms. Where we took different evaluation metrics into consideration such as recall and f-measures in order to avoid the problem of over-fitting as the data we have is imbalanced. GP will be compared with other well-known classifiers in a comprehensive study.

This work differs from other related work researches in that the data processed here was not used before in any machine learning prediction research. Moreover, this data set is recent which makes the results more reliable and accurate for current climate. Moreover, GP fitness function was tested using different metrics.

3. PROPOSED APPROACH

The reason GP was chosen as our approach is that the weather is a dynamic and nonlinear process and GP is a flexible algorithm and uses variable length of its programs thus it can be used for weather prediction. Also, in GP accurate results can be achieved without the need of any analytical knowledge of the solution. Furthermore, GP programs are implemented recursively because they are flexible and can be represented in

ISSN 2319-7595

International Journal of Information Systems and Computer Sciences, Vol.7. No.1, Pages : 01-07 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJISCS/static/pdf/Issue/icsic2017sp14.pdf

a tree structure of variable size. At last GP was efficiently used in weather detection such as [17]-[18].

GP [12] is an evolutionary computation technique that utilizes the same properties of natural selection found in biological evolution. The mechanism of GP starts with a collection of functions and randomly combines them into programs; then run the programs and see which gives the best results. This process can be summarized into 5 main steps as shown in Figure 1:

1. Generate an initial population of programs in a random way, each program consists of mathematical and logical functions with different variables and constants.
2. Evaluate each program by applying the fitness function on it, which checks how well this program does in order to solve the problem we want. In this paper, the problem we are trying to solve is to build a classification model and the fitness function used in evaluating this model is F-measure as stated in equation (7) and called GP-F. While in traditional approached they used accuracy of the model as a fitness function, which we implemented here and called it GP-ACC.
3. Select two programs and consider them as parents using one of the selection methods proposed in the literature such as Roulette wheel selection, Tournament selection, Rank selection …etc.
4. Generate new programs (children) from selected parents by applying crossover and mutation on them. Where the crossover and mutation are applied based on their probabilities set in the GP settings.
5. Consider the new children as new population and repeat the process from step 2.

This process is repeated until reaching termination criteria which can be finding the best program or reaching the maximum number of generations. Although the GP training process takes long time to build the best model, the generated model is very fast to apply since it is a program.

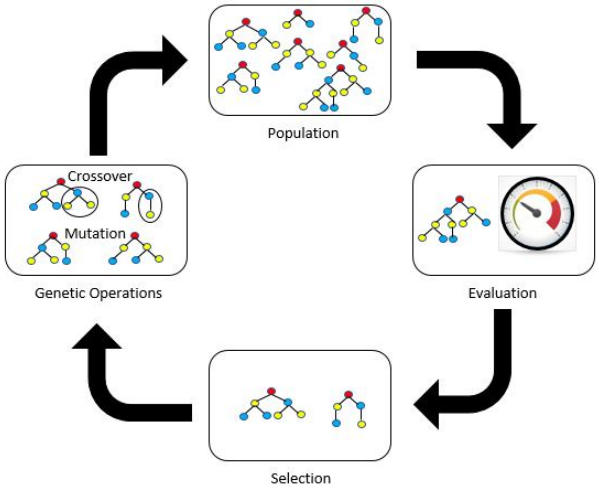

Figure 1: GP Process

4. EXPERIMENTS AND RESULTS

The performance of proposed approach was evaluated by conducting a number of experiments.

4.1 Data

The Metar and SYNOP (Surface Synoptic observation) data file of Lake Charles airport (LCH) was provided to us by ArabiaWeather [14]. The data contains 104 different attributes and 14,969 instances recorded from 21st of December 2015 to 13th of November 2016 for the area of Lake Charles in Louisiana State [15].

4.2 Data preprocessing

The preprocessing step is the most vital step in machine learning, which is complicated merged steps of data cleaning, data reduction and data replacement. Filling the incomplete data and removing the noisy and inconsistency of the data is called data cleaning. Reducing the volume of data by taking a certain group of months that the thunderstorm occurs in the most which is called the data reduction step. Finally replacing the nominal data with numeric data that represents it as the data replacement step.

The missing data percentage was reduced in the main attributes form (2% to 98%) to 0% by applying the following:

1. Duplicating the data to fill the missing values when there is a change measured by change of time.
2. Taking average of the records before and after the missing record and place it in the missing field.
3. Combining multiple columns that are connected in the meaning in one column.
4. Filling missing data based on other attributes or columns values.

ISSN 2319-7595

International Journal of Information Systems and Computer Sciences, Vol.7. No.1, Pages : 01-07 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJISCS/static/pdf/Issue/icsic2017sp14.pdf

Figure 2 represents the thunderstorm occurrences in reference to the period of time that starts from 21st of December 2015 to 13th of November 2016. Where 1 value represents thunderstorm occurrence and 0 value represents the nonoccurrence of thunderstorm. Figure 2 also indicates that the highest rates of thunderstorms are distributed in the dates between May 2016 and September 2016. Which led us to minimize the training data to cover a smaller period of time in order to prevent the problem of over-fitting.

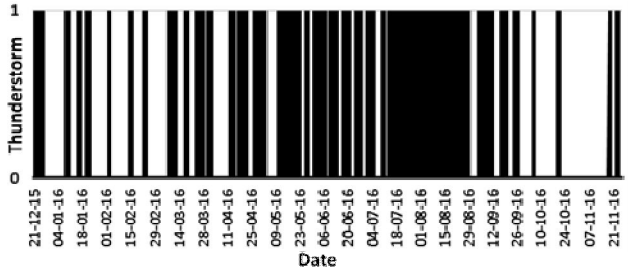

Figure 2: Thunderstorms vs. Time for one year

The number of occurrences of thunderstorms all year was 597 occurrences. When analyzing the data for each month (May-September) we decided to consider the data of the summer season (June, July, and August) of LCH, knowing that it has a total of 278 thunderstorm occurrences which is approximately half of the overall thunderstorm occurrences.

The result of data preprocessing is a data file that contains 42 different preprocessed attributes and 3,910 instances that the Metar and SYNOP recorded for the period between June, 2016 to August 2016; for the area of Lake Charles in Louisiana State.

4.3 Experiments settings

The experiments we did were as the following. First, feature selection -which is also known as variable selection- was applied on the data to take the most relevant features that are most related to thunderstorms to use in the development of our model. The feature selection process was performed using WEKA [10]. The total number of attributes after preprocessing was 44, and after feature selection we got only 31 features plus one feature for class label. The feature selection method used is based on gain ratio, which is the ratio of information gain to the essential information, it is biased towards attributes with a large number of values, by taking the attribute with the highest gain ratio and using it as the splitting attribute thus the total number of attributes is reduced.

Equation (1) is used to calculate the expected information where T is the training data and |T| is the total number of records, r represents a specific value for each feature and the Freq is all the possible values in T feature, where i goes from 1 to n (maximum number of possible values) [11].

$$I(T) = -\sum_{i=1}^{n} \frac{freq(ri.T)}{|T|} \times log2\left(\frac{freq(ri.T)}{|T|}\right) \quad (1)$$

Equation (2) [11] calculates the essential information for a specific value of split (S) where Tj represents the attribute data of number j with m possible attributes.

$$IS(T) = \sum_{j=1}^{m} \frac{|Tj|}{|T|} \times I(Tj) \quad (2)$$

Equation (3) [11] gives the value of information gain of split (S).

$$G(S) = I(T) - IS(T) \quad (3)$$

Equation (4) [11] calculates the information gain ratio between the information gain and the essential information.

$$GR(S) = \frac{G(S)}{IS(S)} \quad (4)$$

The second step was to split the data into two sets 66% for training and 34% for testing. Table 1 shows the data records distribution after splitting it for training and testing. Training set is used in training the GP, while testing set is used to test the goodness of the produced model in predicting new data (thunderstorms).

Table 1: Data after splitting

	Total Records	Thunderstorms
Testing set	1330	93
Training set	2579	185

GP experiments were performed using the Java Genetic Algorithms Package (JGAP) [16] using the settings shown in Table 2. The classifiers used in the comparison with GP results are implemented in WEKA [10], and have the following settings:

- **Bayes Network (BN):** Learning algorithm using various search algorithms and quality measures. The estimator algorithm for finding the conditional probability tables alpha was set to 0.5, and the search algorithm used was hill climbing.
- **Naive Bayes (NB):** Class for a classifier using estimator classes. The batch size was set to 100.
- **J48:** Class for generating a pruned or unpruned C4.5 decision tree. One fold was used for pruning, two folds were used for growing the tree, and the minimum number of instances per leaf was 2.
- **IBK:** K-nearest neighbors' classifier, which can select an appropriate value of K based on cross-validation and also performs distance weighting. The number of neighbors was

ISSN 2319-7595

International Journal of Information Systems and Computer Sciences, Vol.7. No.1, Pages : 01-07 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJISCS/static/pdf/Issue/icsic2017sp14.pdf

one, brute force search algorithm was used for nearest neighbor search, and the window size was set to 0.

- **Multilayer perceptron (MLP):** neural network classifier that uses back propagation to classify instances. The learning rate was set to 0.3, the momentum to 0.2, and the number of hidden layers was (attributes + classes) / 2.
- **K*:** An instance-based classifier, that is, the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function. Global blending was set to 20.
- **JRip:** Implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER). One fold was used for pruning, two folds were used for growing the tree, the number of optimization runs was 2, and the minimum total weight of the instances in a rule was 2.0.
- **SMO:** Implement John Platt's sequential minimal optimization algorithm for training a support vector classifier. The complexity parameter C was set to 1, the calibration method used was multinomial logistic regression.

Table 2: GP settings

Population Size	1000
Number of generations	1000
Maximum Initial Depth	5
Maximum Crossover Depth	8
Mutation Probability	0.1
Crossover Probability	0.5
Tournament Selector	4
Function Probability	0.7
New Chromosomes Percentage	0.2
Function list	21 mathematical and logical functions

4.4 Results

After preprocessing the data, and building our proposed model, and to evaluate it well we need to compare it with other classifiers. Therefore, we applied 8 well known classifiers on the data using the same training and testing sets.

Table 3 shows results of applying some of the classifiers algorithms that WEKA provides on the data. The four columns summarize the confusion matrix results, Where:

– True Positive rate (TPR): means the rate of which there is a thunderstorm that is correctly detected. (Higher values are the better)

– False Negative Rate (FNR): the rate of which there is a thunderstorm but it is classified as no thunderstorm. (Lower values are better)

– False Positive Rate (FPR): the rate of which there is no thunderstorm but it is classified as thunderstorm. (It can be seen as a false alarm)

– True Negative Rate (TNR): the rate of which there is no thunderstorm and it is classified as no thunderstorm.

Table 3: Confusion Matrix of GP-F and other classifiers

Classifier	TPR	FNR	FPR	TNR
BN	58.06%	41.94%	15.84%	84.16%
NB	72.04%	27.96%	15.60%	84.40%
J48	30.10%	69.90%	1.37%	98.63%
IBK	43.01%	56.99%	4.28%	95.72%
MLP	38.70%	61.30%	3.31%	96.69%
K*	38.70%	61.30%	3.15%	96.85%
JRip	32.25%	67.75%	1.53%	98.47%
SMO	29.03%	70.97%	1.05%	98.95%
GP-ACC	17.16%	82.83%	0.461%	99.68%
GP-F	**77.31%**	**22.69%**	29.06%	70.94%
(Best run)	(88.17%)	**(11.83%)**	(19.16%)	(80.8%)

It can be seen from Table 3, that GP-F has the highest TPR (77.31% as an average of 30 runs, where the best TPR value was 88.17%). Moreover, GP-F got the lowest FNR value (22.69%), and a higher TPR and a lower FNR than GP-ACC. From Table 3 we can calculate many metrics that are used to evaluate classifier's efficiency, such as accuracy, recall, ROC area, F-measure.

Accuracy is calculated using Equation (5), which is the number of TP plus TN divided by the overall population (TP+TN+FP+FN). Where TP is the true positive which means correctly detecting a thunderstorm. TN is true negative that means correctly detecting a normal day without thunderstorms. FP is false positive which is a normal day classified as a day with thunderstorm. FN is false negative where a day with thunderstorms is classified as a normal day.

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP} \quad (5)$$

While recall (also known as sensitivity) as shown in equation (6) is the fraction of the true positives that shows there is Thunderstorms and it is detected (TP) to the overall of true positive and false negative cases in the data.

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

F-measure as shown in equation (7) is the value that combines precision and recall by multiplying 2 with the division of the multiplication of them over their sum.

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

Where Precision is calculated as in Equation (8):

$$Precision = \frac{TP}{TP + FP} \quad (8)$$

ISSN 2319-7595

International Journal of Information Systems and Computer Sciences, Vol.7. No.1, Pages : 01-07 (2017)
Special Issue of ICSIC 2017 - Held during 23-24 September 2017 in Amman Arab University, Amman-Jordan
http://www.warse.org/IJISCS/static/pdf/Issue/icsic2017sp14.pdf

AUC is calculated as in Equation (9) and it is defined by the area under the ROC curve which is represented by the true positive rate to the false positive rate at various thresholds.

$$AUC = \frac{(1 - FPR) \times (1 + TPR)}{2} + \frac{FPR \times TPR}{2} \qquad (9)$$

Table 4: Results of GP-F (average of 30 runs) with other classifiers

Classifier	Accuracy	Recall	Precision	AUC	Fmeasure
BN	82.33%	0.581	0.2160	0.71	0.315
NB	83.53%	0.720	0.2577	**0.78**	0.380
J48	93.83%	0.301	0.6222	0.64	0.406
IBK	92.03%	0.430	0.4301	0.69	0.43
MLP	92.63%	0.387	0.4675	0.68	0.424
K*	92.78%	0.387	0.4800	0.68	0.429
JRip	93.83%	0.323	0.6122	0.65	0.423
SMO	**94.06%**	0.290	0.6750	0.64	0.406
GP-ACC	93.77%	0.171	**0.892**	0.58	0.287
GP- F	71.39%	**0.773**	0.1671	0.74	**0.74**
(Best run)	(80.45%)	**(0.88)**	(0.23)	**(0.85)**	**(0.78)**

As observed from Table 4, SMO was run using the default settings and it got the highest accuracy value, however it has the lowest thunderstorm detection (TP) which is a good example of over-fitting problem that can easily occur as the data is imbalanced. Whereas, GP-F got an average accuracy of (71.39%), which again does not represent the efficiency of GP as the data is imbalanced. Looking at other metrics such as recall measure where GP-F got 0.773 and in the best run recall value got 0.88 which are higher than regular GP (GP-ACC) and higher than all the other classifiers. For AUC metric NB got the highest value 0.78, where GP-F got a very close value 0.74, which is higher than 8 other classifiers including GP-ACC. Moreover, GP-F got the highest AUC value in one of the 30 runs which was 0.85, which is the highest compared with all the other classifiers, also it got a higher AUC average (0.74) than GP-ACC. GP's F-measure value is the highest too in the table getting a value 0.74 in average and 0.78 in its best run.

As a summary of GP's results, GP got the best TPR and FNR values, and best recall and f-measure. Where it also got competitive values in accuracy and AUC.

Figure 3 shows a tree representation of the best generated GP program from one run of the 30 runs, the figure illustrates the LISP representation of the following program:

(cosine (X9 * (1.0 + X11) * X17)) / (cosine (X17 * 1.0 * ((cosine X12) * X12 * (X6 + X16))))

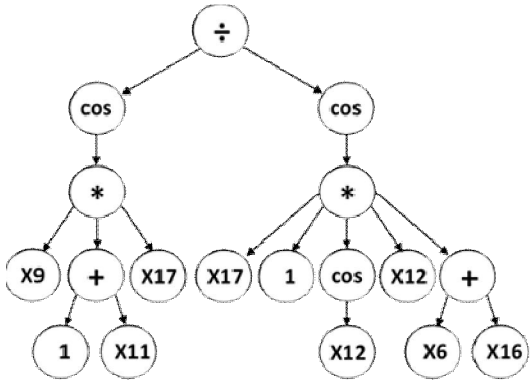

Figure 3: GP Best generated program

5. Conclusion

This paper goal was to use GP to build a model that can predict thunderstorms. The data used in this paper is a real data related to Lake Charles airport, which was collected recently from December 2015 to November 2016. Data preprocessing step was done, including data cleaning, data reduction and data replacement. Then, feature selection process was applied. After that, we applied GP and compared its results with eight other well-known classifiers. The data used was divided into training (66%) and testing (34%).

The Proposed approach used Fmeasure as a fitness function (GP-F) in contrast to traditional GP approach that uses accuracy as a fitness function (GP-ACC). GP-F was tested using five metrics: accuracy, recall, f-measure, AUC. The results showed that GP-F got the highest detection rate of thunderstorms compared with 8 other classifiers, and got highest values of Recall, and F-measure and competitive values in accuracy and AUC metrics.

Our future work includes trying other ways of training GP on the data, which can be done by solving the unbalancing feature of the data. Also, we are interested in testing other classifiers on the data and comparing the results.

Acknowledgments

This research was partially supported by ArabiaWeather. We thank our colleagues from ArabiaWeather who provided insight, data and expertise that greatly assisted the research. We thank Mr. Yazan Dabain, head of weather technology department in ArabiaWeather for assistance with understanding the format of the data, and Mr. Nasser Haddad Meteorologist in ArabiaWeather for helping in understanding the weather forecasting and thunderstorms scenario.

References

1. Shaminder Singh, Pankaj Bhambri and Jasmeen Gill, **Time series based temperature prediction using back propagation with genetic algorithm technique**, IJCSI, Volume 8, PP.28-32, September 2011.

2. Pooja Malik, Prof. Saranjeet Singh, Binni Arora, **An Effective Weather Forecasting Using Neural Network**, IJEERT, Volume 2, PP 209-212, May 2014.
3. Aditya Grover, Ashish Kapoor, Eric Horvitz, **A Deep Hybrid Model for Weather Forecasting**, ACM, PP.1-8, August, 2015.
4. J. Williams, D. Ahijevych, C. Kessinger, T. Saxen, M. Steiner and S. Dettling, **A Machine Learning Approach to Finding Weather Regimes and Skillful Predictor Combinations for Short-Term Storm Forecasting**, Federal Aviation Administration, V1.4, PP.1-6, 2008.
5. Daniel Bejarano, Adriao Quiroga, **Wind Prediction: Physical model improvement through support vector regression**, Stanford University, PP.1-6, December 2013.
6. Waylon Collins, and Philippe Tissot, **An artificial neural network model to predict thunderstorms within 400km2 South Texas domains**, RMetS, PP. 650–665, April 2015.
7. Waylon G. Collins and Philippe Tissot, **Thunderstorm Predictions Using Artificial Neural Networks**, INTECH, Ch. 10, PP.2-31, October 19, 2016.
8. BBC news, **Does bad weather cause plane crashes**, BBC.com/news, AirAsia QZ8501, December, 2014.
9. Juan R. Rabunal, Julian Dorado, **Artificial Neural Networks in Real-Life Applications**, London: IDEA group publishing, PP. 353, 2006.
10. WEKA, version: 3.8, Retrieved on: September 5, 2016, from: http://www.cs.waikato.ac.nz.
11. SAS, **Visual Analytics 7.2 User's Guide**, USA: SAS Institute Inc., Ch. 37, PP.281, 2015.
12. John Koza, **Genetic Programming: On the Programming of Computers by Means of Natural Selection**, Massachusetts Institute of Technology, Ch. 1, 1992.
13. Margaret Rouse, **AWS analytics tools help make sense of big data**, whatis, February 2016.
14. ArabiaWeather Inc., www.arabiaweather.com, located at business park, Amman, Jordan.
15. National Oceanic and Atmospheric Administration NOAA, Retrieved on: September 29, 2016, from: data.noaa.gov/.
16. Java Genetic Algorithms Package (JGAP), source code, Retrieved on: January 2012, from: jgap.sourceforge.net.
17. HaiBo Hu, **Rainstorm flash flood risk assessment using genetic programming: a case study of risk zoning in Beijing**, Natural Hazards, Vol. 83, pp. 485-500, 2016.
18. Giovanna Martínez-Arellano and Nolle, Lars, **Genetic Programming for Wind Power Forecasting and Ramp Detection**, The 33rd SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, PP. 403-417, 2013.