EAEE E9305 – Earth and Environmental Engineering
Master's Research

Application of Graph Neural Networks on Prediction of Wildfire Evolution

Project Report

Submitted by:

Tianxiao Shen(ts3326)

Faculty Advisor:

Prof. Pierre Gentine

https://github.com/Stx980212/GNN_Wildfire

Aug 2022

Columbia University in the City of New York

New York, New York

1. Introduction

As climate change intensifies, the damage caused by wildfires is becoming a concern. Therefore, predicting wildfire's emergence and spread is critical to controlling wildfire damage. Many studies have conducted predictive modeling of wildfires. Still, these models have mainly focused on predicting the probability of wildfire emergence and have rarely oriented towards wildfires that already exist to predict their spatial-temporal evolution.

Apart from theory-driven physical and empirical models, data-driven machine learning models have been applied to this field. For example, researchers have used ConvLSTM in modeling wildfire dynamics (Burge et al., 2020). Compared to computational-expensive physical models and case-sensitive empirical models, ML models are able to learn the evolution patterns of wildfire through variables including land cover, elevation, wind, temperature, etc., and once trained, could produce more accurate predictions based on the patterns learned in a more efficient way.

Besides Convolutional Neural Networks (CNN), Graph Convolutional Neural Networks (GCN) are also used to solve spatial-temporal problems. T-GCN-LSTM (Liu et al., 2019) and ST-GCN (Yu et al., 2018) have been developed to predict traffic volume in the city. Instead of assuming the input data is continuous in Euclidean space like CNN, GCN is more flexible with multiple scaling data and non-Euclidean data, which could be potentially valuable for predicting wildfire evolution.

This project is a master's research project aimed at discovering the effect of applying GCN and its variants to predict the evolution of wildfire, based on a previous course project on ConvLSTM prediction for wildfire. In this project, the same datasets with necessary reconstruction are input into a variety of spatial-temporal models to discover the advantages and disadvantages of CNN and GCN models in wildfire evolution. Similar parameters are used according to the scenario with optimal results in the previous project. Training performance, including testing loss and time elapsed, will be collected through repetitive training, and conclusions will be made by comparing the performances of different models.

2. Models

2.1. ConvLSTM

ConvLSTM uses CNN and LSTM as algorithms to learn the spatial and temporal behavior of the input data, respectively. As an improvement of the previously proposed CNN-LSTM, which uses the output of CNN layers as the input of LSTM, ConvLSTM makes a better combination of CNN and LSTM by embedding convolutional operations into the input-to-state and state-to-state transitions of LSTM cells, forming a ConvLSTM block.

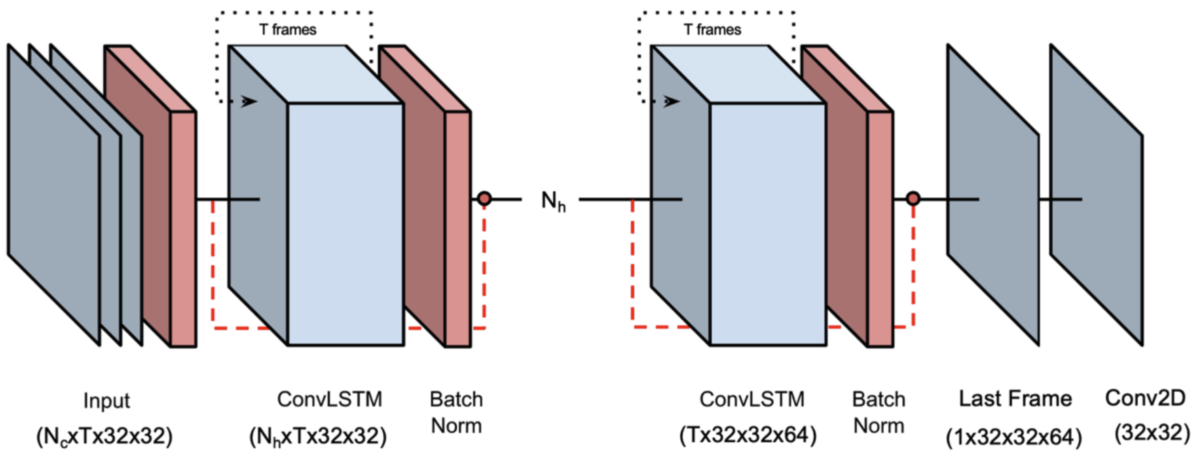


Fig 1. Architecture of ConvLSTM (Burge et al., 2020)

The whole architecture of ConvLSTM is shown in Fig 1, where input datasets with the feature number N_c and sequence length T are input into the ConvLSTM. After N ConvLSTM blocks with Nh kernels for each CNN layer, the input data is processed as a prediction of the wildfire spread at the next time step.

2.2. GConvLSTM

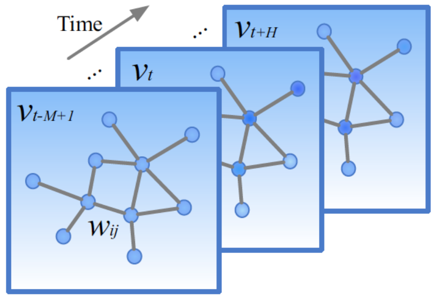


Fig 2. Graph structured data (Yu et al., 2019)

The structure of graph spatial-temporal data is shown in Fig 2. Similar to pixels and kernels for CNN, GCN uses points and edges to represent the features of objects and the adjacency between objects while providing a more flexible mechanism for multi-scale datasets due to its fractal nature.

In this project, GConvLSTM is proposed to make a direct comparison between the performance of GCN and CNN in this case. With the same architecture as ConvLSTM in Fig 1, all CNN layers in ConvLSTM are substituted with GCN layers to form the GConvLSTM.

2.3. Temporal-GCN-LSTM (T-GCN-LSTM)

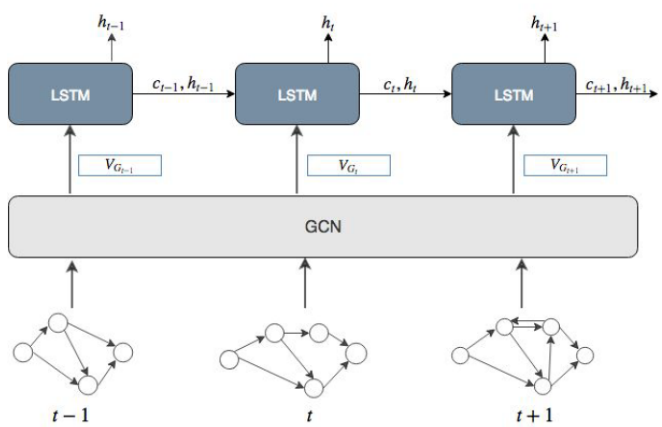


Fig 3. Architecture of Temporal-GConvLSTM (Liu et al., 2019)

T-GCN-LSTM was designed in past research as a part of a complicated forecasting model for traffic volume production (Liu et al., 2019). The architecture is shown in Fig 3. The datasets are input into a 2-layer classification GCN, then the output of the GCN layer is input into an LSTM layer for the temporal dimension.

2.4. Spatial-Temporal-GCN (ST-GCN)

Unlike other models above, ST-GCN uses a gated 1-D CNN rather than LSTM to learn the temporal pattern from the input data. It is reported that ST-GCN has a significantly higher computational efficiency than other models with LSTM layers for the temporal aspect (Yu et al., 2018), which could potentially depict a good performance in wildfire prediction.

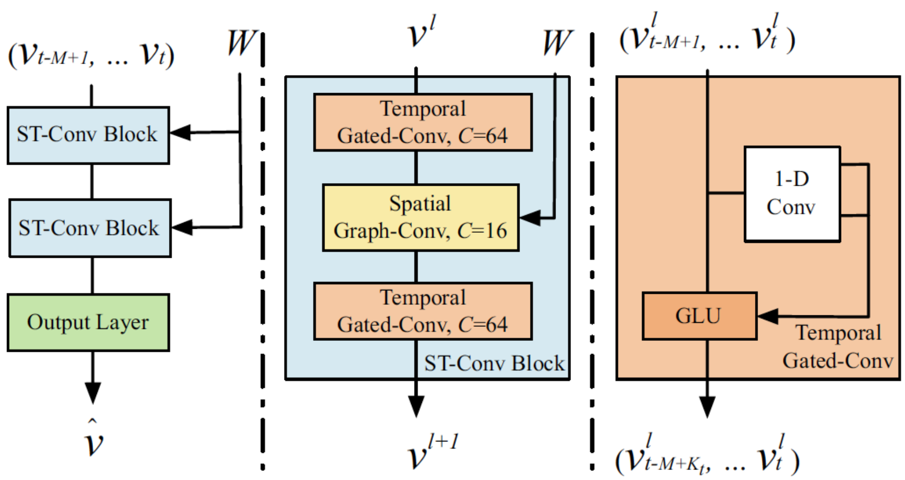


Fig 4. Architecture of ST-GCN (Yu et al., 2018)

The architecture of ST-GCN is shown in Fig 4. An ST-GCN model consists of two spatial-temporal convolutional blocks (ST-Conv blocks) and an output layer in the end. Each ST-Conv block contains two temporal gated CNN layers and one spatial GCN layer in the middle. The residual connection and bottleneck strategy are applied inside each block. The input v_{t-M+1}, ..., v_t is uniformly processed by ST-Conv blocks to explore spatial and temporal dependencies coherently. Comprehensive features are integrated by an output layer to generate the final prediction.

3. Data and Method

3.1. Datasets

Data Name	Spatial Resolution	Temporal Resolution	Unit	Source
Wildfire Evolution	**500 m**	**24 hours**	**-**	**FIRED Product**
Terrian Property				
Land Cover	500 m	-	-	MODIS MCD12Q1
Vegetation	250 m	-	-	MODIS MOD13Q1
Elevation	450 m	-	m	GMTED 2010
Meteorological Data				
10m u-component Wind Speed	-	24 hours	m/s	ERA5
10m v-component Wind Speed	-	24 hours	m/s	ERA5
2m Temperature	-	24 hours	°C	ERA5

Table 1. Details of the input dataset.

3.1.1. Objective Variable: Wildfire Evolution

The evolution of fire during wildfire events is obtained from the Fire Delineation (FIRED) product, which derives the spread of wildfire from the MODIS MCD61A1 Burned Area product. FIRED uses an algorithm to transform the continuous observation of wildfire occurrence on a continent-scale into individual fire events with spatial-temporal grids representing the burning area on each day during the wildfire event. Similar to the previous project, in order to make a more reasonable reconstruction of the spread of wildfire, only fire events that 1) occur in the US region, 2) have a total burned area between 10 km^2 and 30 km^2, and 3) have a duration more than three days. The spatial distribution is represented by 32*32 pixels, with a resolution of 500 meters, and the temporal resolution is 24 hours. The input dataset includes 5074 fire events with an average duration of 9 days.

3.1.2 Other Input Variables

For land cover type and vegetation availability, the MODIS Land Cover Type (MCD12Q1) product and MODIS NDVI (MOD13Q1) product are used, with a resolution of 500 m and 200 m respectively. The 15-arc second Global Multi-resolution Terrain

Elevation Data 2010 (GMTED 2010) is used to take the terrain elevation into consideration during the training.

Along with the terrain property, three meteorological variables, including 10m u,v-component of wind speed and 2m temperature, are retrieved from the ERA5 datasets. The input dataset is at a daily scale, and these meteorological data for each day during the event is an average of the collected data at 2 am, 8 am, 2 pm, and 8 pm local time at the event location centroid.

3.1.3 Data Overview

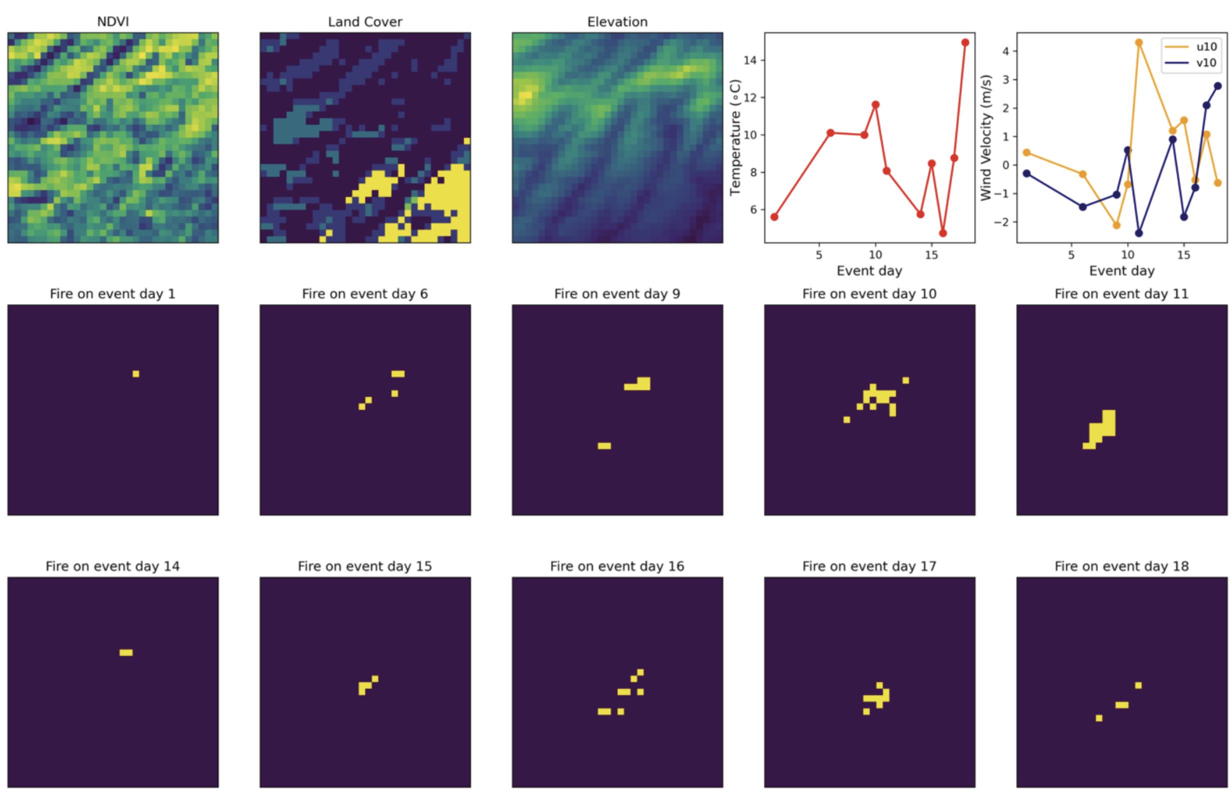


Fig 5. An example of data during a fire event. (Fang and Juang)

Fig 5 shows an example of input data of a fire event. Discontinuous dates between adjacent wildfire images indicate missing data during the wildfire event.

3.2. Adjacency Matrix: Converting raster data to graph relationship

Instead of getting the adjacency information from the column and row orders of the input rasterized dataset like CNN, GCN does not include such information in the form of the input dataset; thus, an additional adjacency matrix is needed to establish the edges in the graph, which indicate the relationship between points.

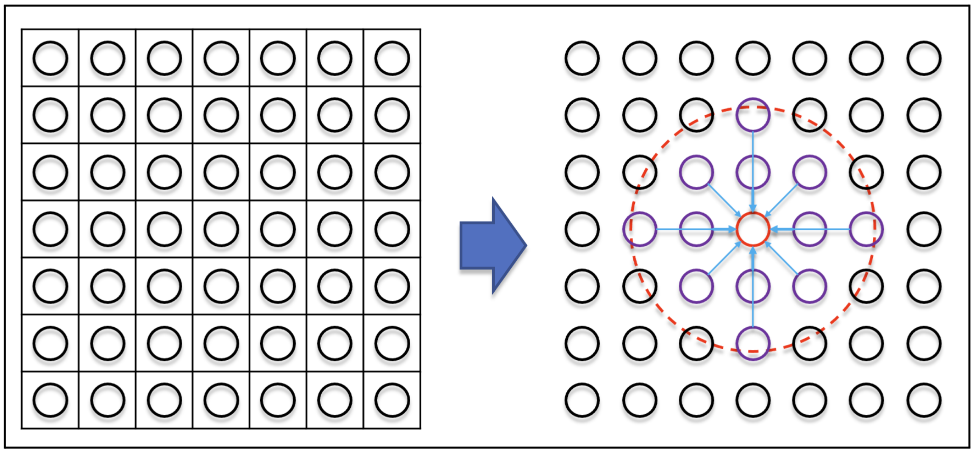


Fig 6. Diagram for setting weighted adjacency matrix.

Fig 6 shows the process of building the adjacency matrix from rasterized data in this case. A point in its center represents every grid in the 32*32 raster. For each datapoint, its adjacency with other points is only considered within a given distance threshold (red dotted circle in the figure) due to the computational efficiency. Also, to accommodate the effect of different distances between the points, the weight matrix is calculated inversely proportional to the distance by the formula below,

$$W_{i,j} = \frac{1}{1 + D_{i,j}}$$

where $D_{i,j}$ is the distance between two points.

3.3. Training Procedure

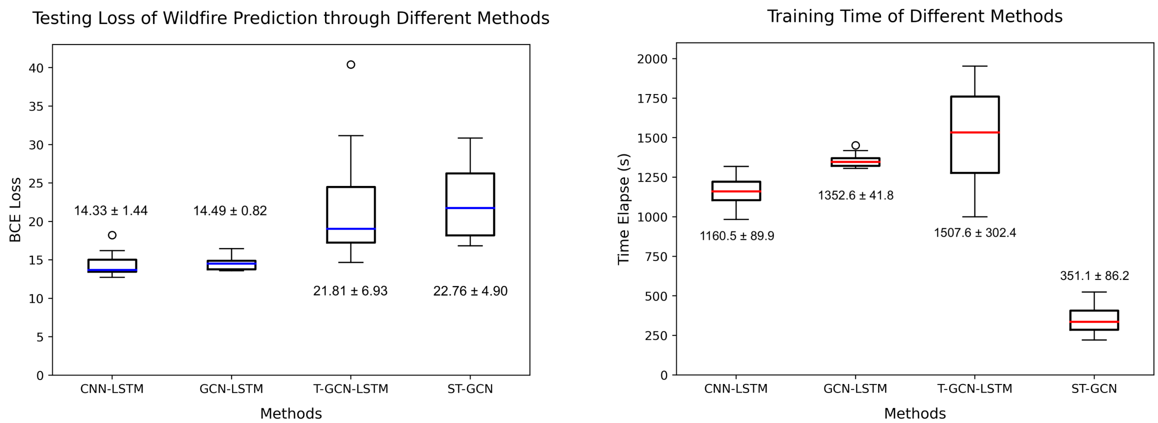
Similar to the former project, four different models were trained, with 80% of the dataset used as training data, 16% of the dataset used as validation data, and 4% of the dataset used as testing data. At each timestep T in a wildfire event N, all variables in the previous timesteps N_0, N_1, …, N_{T-1} were used as input variables to predict the possibility distribution of wildfire in the current timestep N_T. Binary Cross Entropy (BCE) was used as the loss function in the training process.

To make training results comparable across different methods, similar training parameters are used according to the optimal result from the former research. The learning rate and the number of layers were set to be $1*10^4$ and 3, respectively. For CNN layers in training, the size and number of kernels were 3*3 and 32; for GCN layers, the distance threshold of the adjacency matrix was set to be 2.

In order to eliminate the random factor in the training result, the whole process, including the segmentation of data and training of the model, is repeated 15 times in each case.

4. Results

The performances of the trained models are shown in Fig 7(a) below, where the testing losses are presented in a boxplot. Among all the methods concerned, ConvLSTM achieved the lowest average testing loss of 14.33, presenting an obvious advantage over methods in this case. Simply substituting the CNN layers with GCN layers in ConvLSTM did not cause any significant change in the model performance, as shown in the result of GConvLSTM, which indicates the fact that GCN may function in a similar way as CNN. The average testing loss of T-GCN-LSTM went high due to the existence of more extreme cases, while it could still achieve a similar performance as ConvLSTM in some attempts. And the prediction of ST-GCN is significantly worse than other methods, making it the least effective model in this case.



(a) (b)

Fig 7. Performance of different models on wildfire data.

Fig 7(b) compares the total training time of each model, i.e., the computational efficiency of different models. In this study, ST-GCN took the least time to complete the training process, demonstrating an obvious advantage in computational efficiency. And for the rest three methods, training time increases with the complexity of the model. At the same time, the variance of T-GCN-LSTM is exceptionally high, which may be caused by the sparsity of the original data.

5. Discussion

The similar performance of ConvLSTM and GConvLSTM indicates not only the similar performance of CNN and GCN in this case but also the similar way they function in Euclidian space. If we look back to the origin of GCN, we can find that GCN was first proposed as an extension of CNN to non-Euclidean space, so it is not surprising that in this case, when all input data were re-rasterized into the same size, CNN and GCN function in similar ways and produce similar results. But still, GCN provides the possibility that, with a more complicated process, it is possible to connect raster datasets of different resolutions with GCN in the future.

Data quality may have limited the performance of T-GCN-LSTM, whose model is more complicated than GConvLSTM but failed to produce better prediction and more steady performance. The large variance in training time may be a sign of poor input data quality. And indeed, as Fig 6 shows, there could be some missing days in FIRED data, depicting a discontinuous spreading mechanism that may cause confusion when input into the model as a continuous sequence.

ST-GCN shows an obvious advantage over other models in terms of computational efficiency. However, due to the static input of landcover and vegetation, it seems important to have a memory of all previously burning areas during a wildfire event; thus, using GCN as the predictor of temporal trend instead of LSTM may not be a wise choice, which produced least accurate and steady result in this study.

In general, according to the result of the previous project (Fig 8), even the best scenario did not produce an apparent prediction for the direction of wildfire propagation. More likely, the model uses the possibility distribution of all fire events in the input dataset as its prediction result for all cases. So, simply applying these model to the input dataset may not gives out an expected successful prediction. And in the future, introducing algorithms and formulas that can reconstruct the mechanism of wildfire evolution to the model or pretraining the model on synthetic data showing a clear mechanism of wildfire evolution could be perspective methods for further improvement.

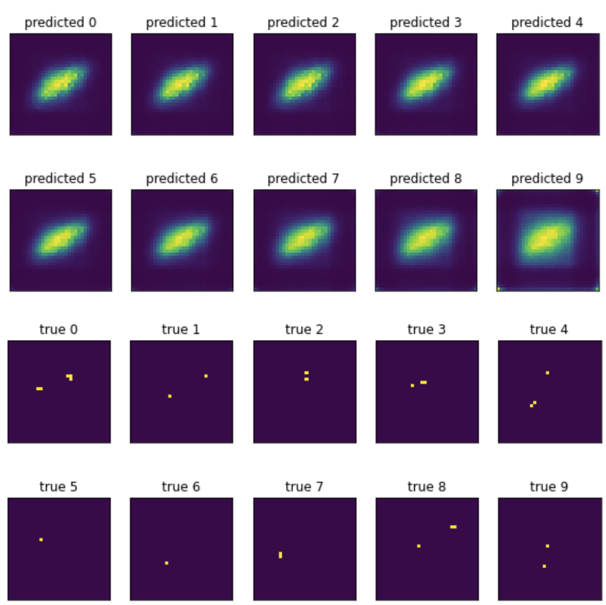


Fig 8. Prediction and actual value in the best scenario of the previous project (Fang and Juang)

Code Availability

The project code is available at: https://github.com/Stx980212/GNN_Wildfire, which includes source code from other open source projects in the references.

Major References

Blach, J. K., St. Denis, L. A., Mahood, A. L., Mietkiewicz, N. P., Williams, T. M., McGlinchy, J., Cook, M. C. (2020). FIRED (Fire Events Delineation): An Open, Flexible Algorithm and Database of US Fire Events Derived from the MODIS Burned Area Product (2001-2019). *Remote Sensing, 2020, 12, 3498; doi:10.3390/rs12213498.*

Burge, J., Bonanni, M., Ihme, M., & Hu, R. L. (2020). Convolutional LSTM Neural Networks for Modeling Wildland Fire Dynamics Running Head: ConvLSTMs for Wildland Fire Dynamics. In arXiv preprint.

Fang, J., Juang, C. (2021). Modeling Short-term Wildfire Spread with Convolutional LSTM: Challenges and Opportunities. *Course Project for Columbia University EAEE4220.*

Liu, Y., Shi, X., Pierce, L., Ren, X. (2019). Characterizing and Forecasting User Engagement with In-app Action Graph: A Case Study of Snapchat. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2023-2031).*

Yu, B., Yin, H., Zhu, Z. (2018). Spatial-Temporal Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*.