HAIL RISK PREDICTION VIA GRAPH NEURAL NETWORKS

A PREPRINT

Ivan Lukyanenko
Department of Control and Applied Mathematics
Moscow Institute of Physics and Technologies
Moscow
lukianenko.ia@phystech.edu

ABSTRACT

Geo-spatial time series is an open area with great potential for theoretical and practical work. In particular, hail risk assessment is necessary to avoid the damage (agriculture, animal husbandry). The aim of the study is to build a model based on graph neural networks. Forecasting has been carried out in the short-term range based on the values of climate variables since 1991. The key features of the problem are: 1) rare events - over the past 30 years there have been less than 700 hail events throughout Russia, 2) the spatial structure of the series data. We are expecting to improve quality of solving such problems by combining methods from [2] and [3].

Keywords Hail risk prediction · GNN · spatial time series

1 Introduction

The main goal of this research is to introduce new GNN architecture by combining state-of-art results [2], [3] in the right way. Our object of research is geo-spatial time series. In common time series is a series of values of a quantity obtained at successive times with equal intervals between them. This research is about more extended concept: spatial time series. Spatial time series are almost the same as ordinary time series, but instead of values we observe some spatial objects. This work provides a solution to the hail forecasting problem. The hails are extreme events, because over the past 30 years there have been less than 700 hail events throughout Russia. The assessment of hail risk is necessary because of its environment damage. According to Verisk's 2021 report [8], in 2020 year losses due to hails reached $14.2 billion in the USA. add citation

The producing architecture is based on works [2], [9] and [3]. [3] introduce **StrGNN** that solves the anomaly detection problem in dynamic graph, this architecture solves big class of problems well, but it should be adapted to narrower tasks. This work produces re-implemented **StrGNN** for our purpose in combination with GNN-network from [2]. [2] based on the model presented in [9]. We are going to use climate data from Google Earth Data, CMIP5, NOAA Storm Events Database, Severe Weather Dataset for training and evaluating our architecture.

2 Problem Statement

Using architecture is described in [3]. This is the parametric family of function to map design space to target space.

2.1 GNN

Graph Neural Networks are used when your data has graph structure. The approach of using graph structure of data and its theoretical proof presented in [5]. The main idea is to aggregate features from neighboring nodes under the assumption of their influence as following:

$$Y = AXW \tag{1}$$

Instead of weighted sum in the layer of neural network there is aggregating of neighboring nodes' features due to dot product with adjacency matrix.

$$Z = f(X, A) = \text{softmax}\big(A \text{ ReLU}\big(AXW^{(0)}\big)W^{(1)}\big) \tag{2}$$

This is an example of 2-layer Graph Neural Network [5].

2.2 StrGNN

Given a temporal network $\{G(t) = \{V(t), E(t)\}_{t=1}^{n}\}$, where $G(t)$ is the graph snapshot at timestamp t consisting of verticles $V(t)$ and edges $E(t)$.
StrGNN works in three stages: ESG, GSFE, TDN.

ESG: Enclosing Subgraph Generation

The goal is to generate enclosing subgraph structure related to the target edge. Employing the whole graph for analysis is computational expensive. Recent work [10] proved that in Graph Neural Networks each node is most influenced by its neighbors. Enclosing subgraph in dynamic graphs definition according to [3] is following:

Definition 1. (Enclosing subgraph in dynamic graphs) For a temporal network $\{G(i) = \{V(i), E(i)\}\}_{i=t-w+1}^{t}$ with window size w, given a target edge e^t with source node x^t and destination node y^t, the h–hop enclosing subgraph $G^h_{x^t,y^t}$ centered on edge e^t is a collection of all subgraphs centered on e^t in the temporal network $\{G(i)^h_{x^t,y^t} | (t - w + 1) \leq i \leq t\}$. Enclosing subgraph contains only topographical information. In case of this, to distinguish the role of each node, nodes should be labeled. According to [3] good labeling contains the information of 1) target node in subgraph; 2) the contribution of each node in target one.
Suggested labeling is following:

$$f(i, x^t, y^t) = 1 + \min(d(i, x^t), d(i, y^t)) + (d_{\text{sum}}/2)[(d_{\text{sum}/2}) + (d_{\text{sum}}\%2) - 1] \tag{3}$$

where $d(i, x^t)$ is the shortest path distance between i and node x^t, $d_{\text{sum}} = d(i, x^t) + d(i, y^t)$.
If $d(i, x^t) = \infty$ or $d(i, y^t) = \infty$, node i is labeled with 0.

GSFE: Graph Structural Feature Extraction
Graph Convolution Neural Network [6] map the subgraph space into embedding space.

TDN: Temporal Detection Network GSFE generates low-dimensional features, but it does not consider the temporal information to determine the class of the target.
Given the extracted feature matrices $\{H_i\}_{i=t-w}^{t} in\mathbb{R}^{K\times d}$, where K is the number of selected nodes

in subgraph and d is dimension of feature for each node.
Employed the Gated Recurrent Units (GRUs) [4] to capture the temporal information as:

$$z_t = \sigma(W_z H_t + U_z h_{t-1} + b_z) \tag{4}$$

$$r_t = \sigma(W_r H_t + U_r h_{t-1} + b_r) \tag{5}$$

$$h_t^{'} = \tanh(W_h H_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{6}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ h_t^{'} \tag{7}$$

where $\circ$ represent the element-wise product, W, U and b are parameters. The GRU network is able to model the future temporal information.

2.3 Loss-function

The criterion is log-loss function. Hail Prediction is considering as classification problem. The output of the model is classified months in the prediction horizon whether hail will appear or not. The problem mathematically summarized as following:

$$w^* = \arg\min_w \Big(-\frac{1}{N} \sum_{t=1}^{N} (y^t \log(g(w, h^t)) + (1 - y^t) \log(1 - g(w, h^t))) \Big) \tag{8}$$

, where w^* is optimal vector of weights of our architecture; w is vector of weights of our architecture; t is timestamp from the beginning of the prediction horizon; N is number of timestamps in prediction horizon; y is target class of the timestamp; g is fully-connected network; h is output of re-implemented version of **StrGNN** [3]

3 Computational Experiment

The experiment goal is to train and evaluate our model on TerraClimate and meteo.ru data.

3.1 Data

3.1.1 TerraClimate

TerraClimate is a dataset of monthly climate and climatic water balance for global terrestrial surfaces. It uses climatically aided interpolation, combining high-spatial resolution climatological normals from the WorldClim dataset, with coarser spatial resolution.

This source provides main data for the experiments. Spatial time series of various climatic variables. We use following variables: mean temperature above the surface, surface pressure, total precipitation, 2 components of wind.

3.1.2 meteo.ru

This data center provides data about hail events from 1991 year in Russia.

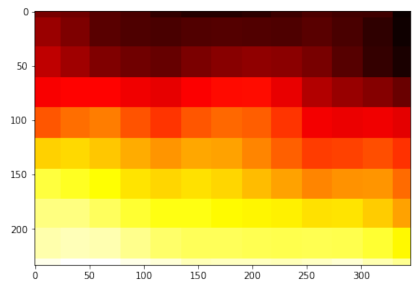
3.2 Experiment Set-Up

Unless otherwise noted, we train re-implemented **StrGNN** described in Section 2.2 and evaluate prediction accuracy, precision, recall on a test year to be predicted. The data for train and test is described in Section 3.1.

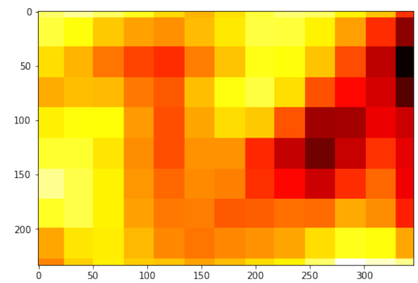
Except of training and evaluating model, the experiment goal is to optimize hyperparameters such as adjancy matrices in Graph-layers. This is important part of research, because it could be interpreted as connections between climatic values.

Provided model is compared with SMOTE + log-regresion. This baseline is described in Section 4.

At every timestamp we have tensor of climatic variables.



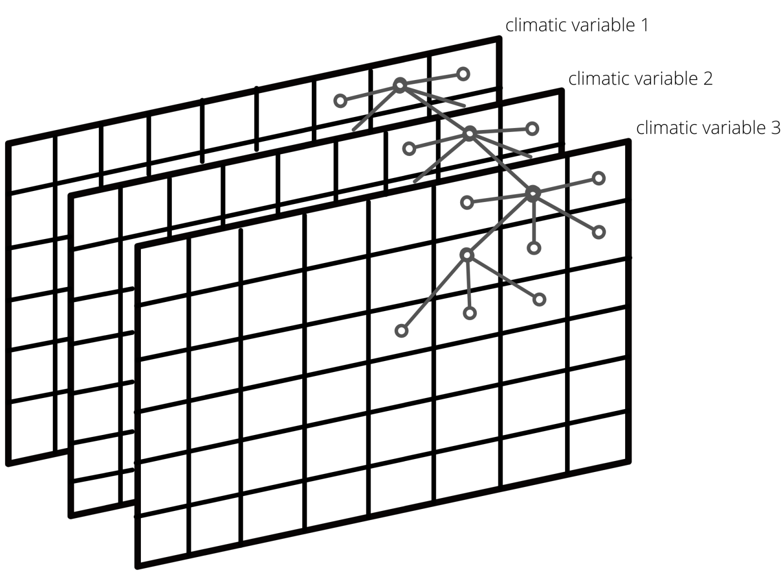
Month mean temperature



Surface pressure

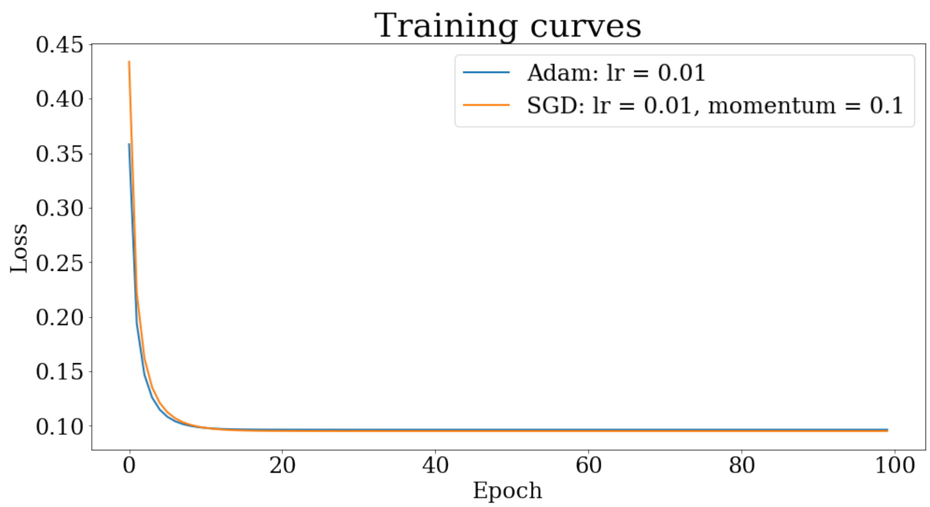
These variables corresponds to Tambov region.

Using GNN-layers it is possible to connect different variables and their spatial neighbourhood. It can be visualized as following:

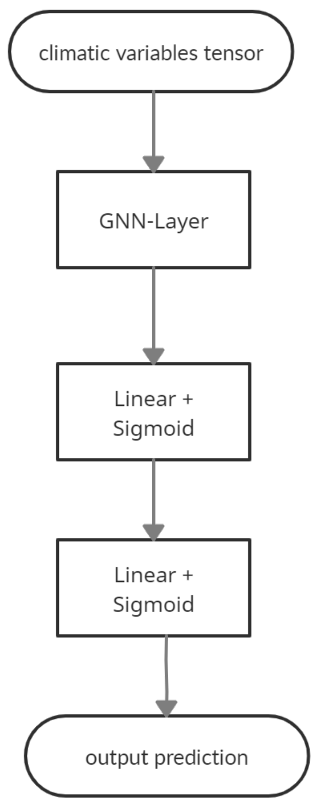


Visualisation of principle

Using only GNN + fully connected network we can achieve good results in classifying timestamps with hail events:



Actual architecture is visualized as following:



HailNet architecture

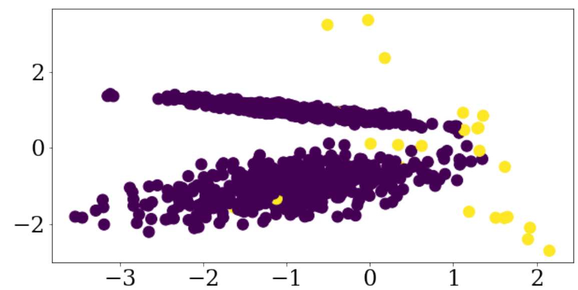
List of Expected tables and figures: temporary

label	precision	recall	f1
0	0.99	0.99	0.99
1	0.99	0.99	0.99

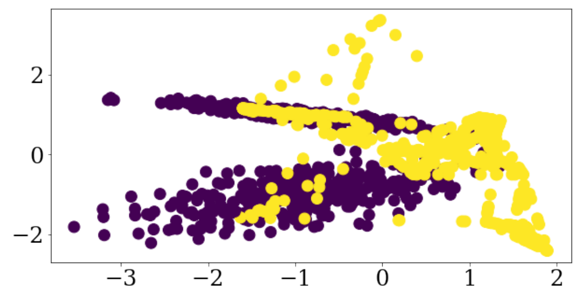
ROC-AUC plot;
plot of predicted probability with vspan labelling

4 Preliminary report

In order to show how does SMOTE works, there is some experiments with baseline SMOTE + logistic regression on synthetic data . Firstly, created synthetic data for classification problem with imbalanced classes in proportions $98/2$.

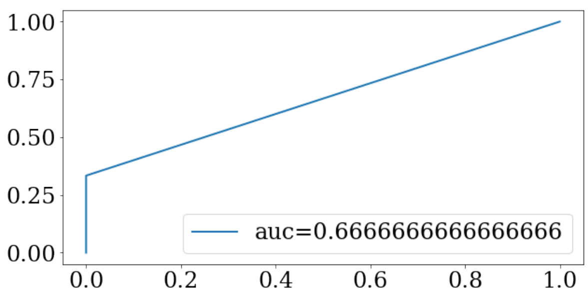


Data distribution before SMOTE

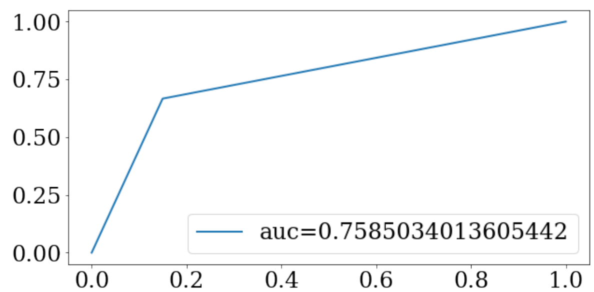


Data distribution after SMOTE

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.



ROC-curve distribution before SMOTE



ROC-curve distribution after SMOTE

ROC-AUC increased. So we consider that SMOTE make our model better in some cases.

5 Baselines

The hail risk prediction is classification problems with imbalanced classes. Hail is extreme climatic event. Using SMOTE we can synthesize new examples for the minority classes [1]. SMOTE with logistic regression is our simplest model. This model we use to compare quality with complex one. Logistic regression maps feature space into 1d probability simplex. $\mathbb{R}^n \to [0,1]$. Logistic regression model is following:

$$y_i = \frac{1}{1 + e^{-x_i w}} \tag{9}$$

where x_i is feature vector, w is parameters of the model, y_i is probability of positive class labeled "1". w find by LBFGS optimization algorithm minimizing log-loss function. The LBFGS optimization algorithm reviewed in details here [7].

References

[1] K. W. Bowyer, N. V. Chawla, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813, 2011.

[2] S. R. Cachay, E. Erickson, A. F. C. Bucker, E. Pokropek, W. Potosnak, S. Osei, and B. Lütjens. Graph neural networks for improved el niño forecasting. *CoRR*, abs/2012.01598, 2020.

[3] L. Cai, Z. Chen, C. Luo, J. Gui, J. Ni, D. Li, and H. Chen. Structural temporal graph neural networks for anomaly detection in dynamic graphs. *CoRR*, abs/2005.07427, 2020.

[4] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[5] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[6] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks, 2017.

[7] A. Mokhtari and A. Ribeiro. Global convergence of online limited memory bfgs, 2014.

[8] Verisk. The hail hazard and its impact on property insurance, 2020.

[9] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks, 2020.

[10] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka. Representation learning on graphs with jumping knowledge networks. *CoRR*, abs/1806.03536, 2018.