HAIL RISK PREDICTION VIA GRAPH NEURAL NETWORKS

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 [9], in 2020 year losses due to hails reached $14.2 billion in the USA.

The producing architecture is based on works [2], [10] 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 [10]. 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 [11] 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_{x^t,y^t}^h centered on edge e^t is a collection of all subgraphs centered on e^t in the temporal network $\{G(i)_{x^t,y^t}^h|(t-w+1) \le i \le t\}$. 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:

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

, 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 Theoretical part

This section contains theoretical assumptions and hypothesis of provided method of forecasting. Due to solving particular problem from climate science, it is necessary to dive into this problem and understand the processes of hail formation.

3.1 How does hail form?

According to [8], hailstones are formed when raindrops are carried upward by thunderstorm updrafts into extremely cold areas of the atmosphere and freeze. Hailstones then grow by colliding with liquid water drops that freeze onto the hailstone's surface. The hail falls when the thunderstorm's updraft can no longer support the weight of the hailstone, which can occur if the stone becomes large enough or the updraft weakens. In most studies, it is noted that most of the growth gradation occurs at a temperature of approximately -10 to -25 ∘C.

Hypothesis 1. *Hail event depends on certain patterns in multivariate time series of climatic variables such as wind vertical component, wind horizontal component, temperature.*

Important ingredient for creating large hailstones is time. Appreciable growth is only attainable if particles remain in an environment conducive to growth for an extended period of time. Some studies suggest that large hailstones spend as much as 10–15 min or more in growth regions of storms.

Hypothesis 2. *Hailstones formation spend approximately 20 min. Hailstones are in the clouds till the updrafts of wind prevent them from falling down.*

3.2 How we can take account of climatic knowledge?

Lemma 1. *Let X - $n \times m$ feature matrix of n objects with n features. Let our objects have graph structure with A - adjacency matrix. Then GNN - layers aggregate features from neighbourhood nodes. Moreover applying GNN layers n times, model aggregates n - step neighbours.*

Proof. First part of the statement:

Adjacency matrix is symmetric matrix and consists of zeros and ones. In the graph layer adjacency matrix operator A operates on X:

$$H_1 = AX \tag{9}$$

$$(H_1)_{i,j} = \sum_{k=1}^{n} A_{i,k} X_{k,i} \tag{10}$$

$X_{k,i}$ represents i feature of k object. $A_{i,k}$ represents connection between i, k objects. Thus, we state that H_1 consists of aggregation rows. i-th row represents aggregation of i-th node neighbours.

Second part of the statement:

Let define raw object as 0-sized neighbourhood. As we proved above after applying A - matrix to zero-sized neighbourhood we achieve matrix of one-sized neighbourhood. This is the base of induction. Let give assumption that on the i-th iteration we have H_{n-1} matrix that represents $(n-1)$-sized neighbourhood. Then after applying matrix A to H_{n-1} one more time we aggregate i - th neighbourhood with i-th node neighbours $(n-1)$-aggregation, in this way we get n-aggregation. Thus, $H_n = A^n X$ contains n-sized neighbourhoods aggregations. □

Due to Hypothesis 1 provided model needs to aggregate information from different climatic variables and its spatial neighbourhood. Relying on Lemma 1 we achieve this aggregation by using GNN layers in our architecture for timestamp embeddings. Due to Hypothesis 1 provided model needs to reveal temporal patterns of multivariate time series for better classification. This temporal structure is analyzed by GRU layers in our architecture. Due to given hypotheses and theoretical justifications, the result architecture is following:
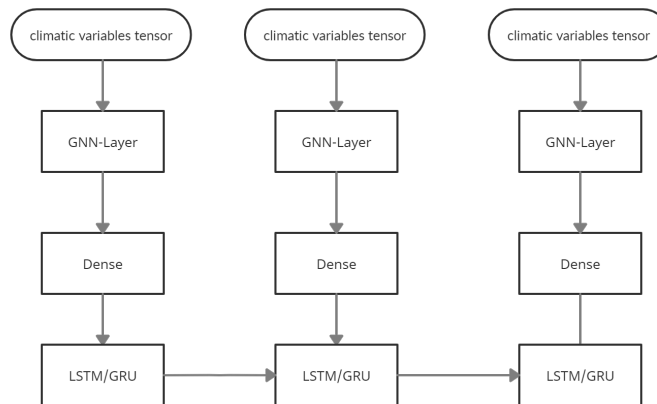
Resulting architecture

4 Computational Experiment

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

4.1 Data

4.1.1 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.

4.1.2 meteo.ru

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

4.2 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.
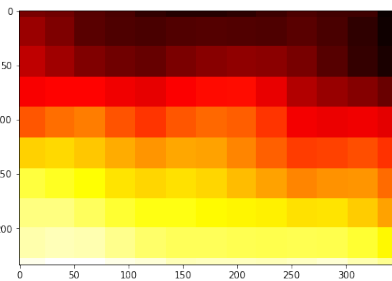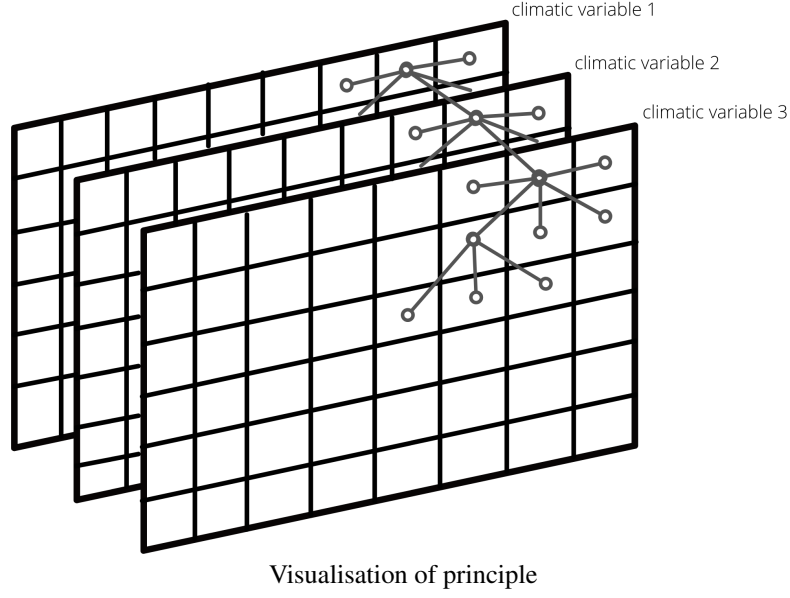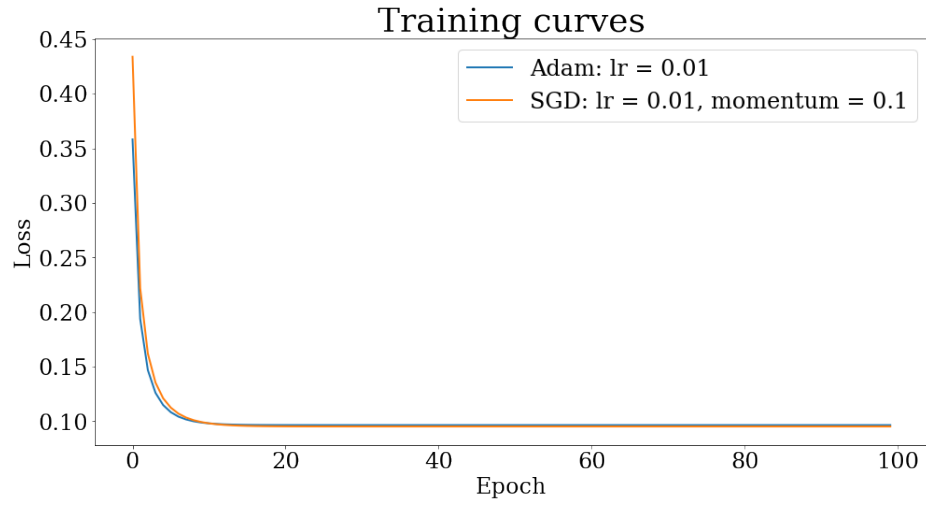

Month mean temperature
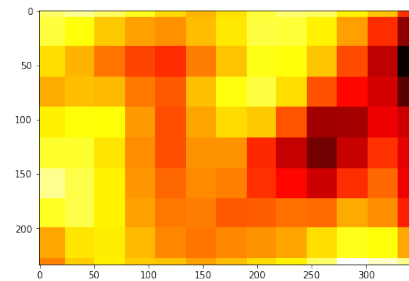
Surface pressure

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:
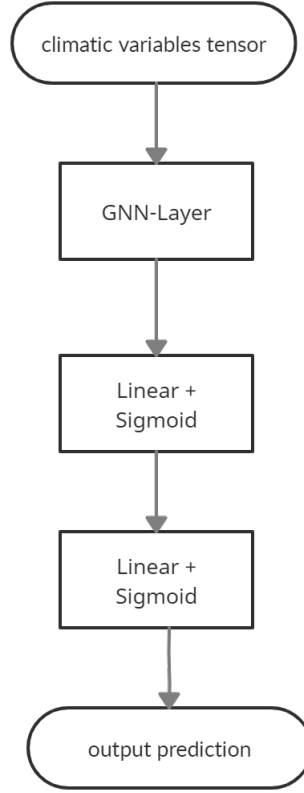
Visualisation of principle

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

4.3 Actual architecture

Actual architecture is visualized as following:

HailNet architecture

In the GNN layer we applying adjacency matrix to climatic tensor to make attention of spatial neighbourhood. After that we applying two linear layers with sigmoid activation function:

$$H_1 = WAX \tag{11}$$

$$H_2 = \sigma(WH_1) \tag{12}$$

$$H_3 = \sigma(WH_2) \tag{13}$$

Looking forward to modify adjacency matrix according to climate knowledge of hail events and add LSTM/GRU layers to forecast future month.

4.4 Experiment flow

1. Creating torch.DataLoader of downloaded climatic data from TerraClimate using Google Earth Engine and hail events data from meteo.ru.

2. Feed 4-sized batches into training algorithm for 100 epochs.

3. Updating HailNet parameters optimizing Binary Cross-entropy Loss with Adam optimizer and SGD optimizer.

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

5   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
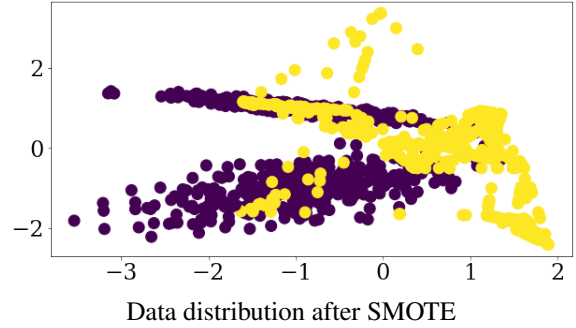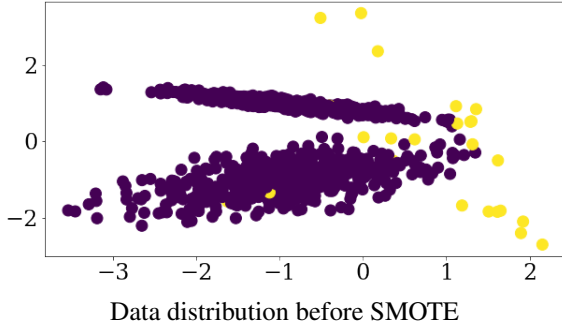
SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

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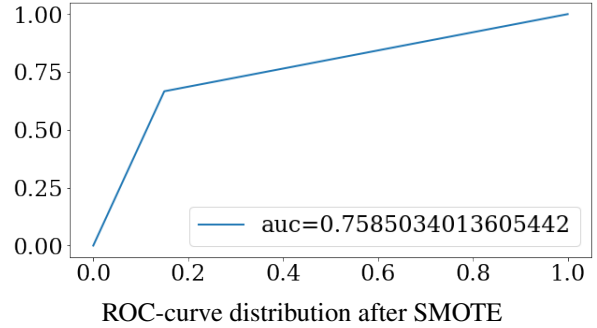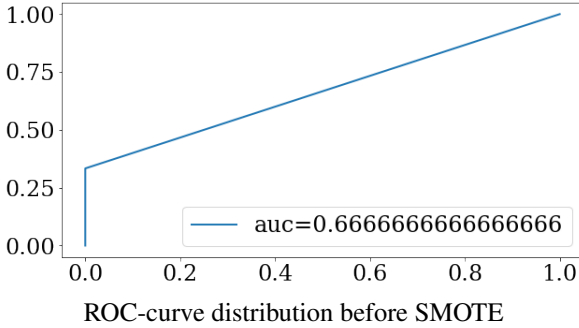
ROC-AUC increased. So we consider that SMOTE make our model better in some cases.

6   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:

$$y_i = \frac{1}{1 + e^{-x_i w}} \tag{14}$$

8

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] NSSL. Nssl's hail research, 2020.

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

[10] 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.

[11] 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.