PSEUDOCODE

|  |
| --- |
| **Algorithm** Granular-based Multi-Layer Spatiotemporal Network |
| 1. **for** 1 to num\_layers **do** 2. Define the calculation of the gates , , , , cell states  and hidden representation  of the spatio-temporal LSTM units according to the equations (8)-(11), (12)-(16); 3. Define fully connected output layers; 4. Initialize the cell/hidden states , ; 5. Initialize the control gate ; 6. Ship the model to the GPU if desired; 7. **end for**   //forward pass   1. **for** 1 to num\_layers **do** 2. Inputs = ; //Define input from training samples; 3. **if** num\_layer ~= 1 **then** 4. Inputs = [Inputs, , ,…]; //Insert spatial information 5. Determine the control gates  according to the granular topology orders; 6. **end if** 7. Inputs = [Inputs, ]; //Insert temporal information 8. Output = nnModel.*forward*(Inputs); 9. **end for** 10. loss = nnModel.*criterion*(Output, y);   //backward pass using Adam optimizer   1. **for** 1 to max\_epoch **do** 2. Initialize gradient to be zeros; 3. **for** time\_steps to 1 **do** 4. nnModel.*backward*(Output, y); **//**backprop from loss 5. **for** num\_layers to 1 **do** 6. Construct layer inputs according to Step 15) to 20); 7. nnModel. *backward*(Inputs, nnModel.weights);   **//**backprop from next\_step   1. Update temporal backprop weights; 2. Update spatial backprop weights; 3. **end for** 4. **end for** 5. **end for** 6. Input test data to generate prediction value; |