This paper presents the application of graph neural networks (GNNs) to the task of node classification. GNNs have been shown to be useful in various classification tasks where data and the relationships between them can be represented using graphs. This research aims to develop a classifier that can identify two possible classes of Twitter nodes: COVID and nonCOVID. COVID nodes refer to Twitter users (nodes) that post tweets related to COVID-19 and nonCOVID are users (nodes) that do not post tweets about COVID-19. For that purpose, in the first step, we implement a pipeline that enables the automatic, continuous collection of data from Twitter and network construction. In the second step, we prepare the data and train a graph convolutional network (GCN) classifier. We compare GCN and multilayer perceptron (MLP) in terms of standard measures: precision, recall, F1 and accuracy. The results show that GCN performs better than MLP in the task of node classification.

Figure 1. Conceptual model of pipeline implemented for collecting and preparing the datasets.

Figure 2. Visual representation of Twitter following network, where node color represents COVID and nonCOVID label in dataset. Links between nodes represent following and friend relations between user nodes, where arrow direction corresponds if user is following or being followed by another node.

Models

Model training consists of two steps. In the first step, node embeddings are created with node2vec method, with dimensions set on 16, 32, and 64. In the second step these embeddings are forwarded to GCN model. Each dimension was created with parameters where the batch size is 64, the learning rate is 0.01, walk length is set to 10, and number of walks is 3 and this was performed in 5 epochs. We trained a model to classify the tweets into one of the two classes defined above. When creating the model architecture and setting the parameters, the ones that gave the best results were selected. The selected parameters are listed below:

- The input vector has the dimension of the embedding vector (16, 32 or 64).
- There are 3 layers with the number of hidden channels is 64 for MLP and 256 for GCN per layer. Dropout is set to 0.3 and learning rate to 0.005.
- Optimizer used for training is Adam.

Table 1. Model precision.

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP + n2v 16d</td>
<td>0.4643</td>
<td>0.2743</td>
<td>0.3449</td>
<td>0.5787</td>
</tr>
<tr>
<td>GCN + n2v 16d</td>
<td>0.7721</td>
<td>0.6029</td>
<td>0.6771</td>
<td>0.8129</td>
</tr>
<tr>
<td>MLP + n2v 32d</td>
<td>0.4825</td>
<td>0.3230</td>
<td>0.3869</td>
<td>0.5962</td>
</tr>
<tr>
<td>GCN + n2v 32d</td>
<td>0.7551</td>
<td>0.6985</td>
<td>0.7721</td>
<td>0.9255</td>
</tr>
<tr>
<td>MLP + n2v 64d</td>
<td>0.4736</td>
<td>0.6217</td>
<td>0.5376</td>
<td>0.6777</td>
</tr>
<tr>
<td>GCN + n2v 64d</td>
<td>0.7916</td>
<td>0.7294</td>
<td>0.7592</td>
<td>0.8137</td>
</tr>
</tbody>
</table>

Table 2. Performance of models in different dimensions.

Tables 1 and 2 show that the GCN model performs better than MLP in all three experiment setups. The best performance is achieved in the case of GCN combined with the embedding dimension set to 64. As expected, higher embedding dimensions provide better results in almost all cases.

Methods and Materials

Dataset All the data is collected from the Twitter social network using a pipeline that we implemented for automatic, continuous collecting of data from Twitter presented in Figure 1. Data is structured so that for each user we have its friends, followers, and a list of published tweets at a certain time Figure 2.

For this study, we collected the Cro-WUS dataset of 8,808 Twitter users from the Republic of Croatia (Cro-ERSUS) and 1,703,626 of their tweets (Tweets dataset) in the Croatian language posted during the fourth wave of the COVID-19 pandemic, distribution of data is shown in Chart 1.

Results

GCN and MLP models were trained using various dimensions of embeddings and compare their performance in terms of accuracy, precision, recall and F1-measure. The Cro-ERSUS dataset of 8,808 Twitter users was split with a ratio (55/30/15) for train, validation, and test. The model was trained on embeddings of different dimensions created using the node2vec algorithm. We experimented with three different embeddings dimensions: 16, 32 and 64. The models were trained on 200 epochs on 100 runs of which the best results are shown in the Table.

According to the results, GCN outperforms MLP in all three experiment setups. The best performance is achieved in the case of GCN combined with the embedding dimension set to 64. As expected, higher embedding dimensions provide better results in almost all cases.

References