P4: Study on Predicting Social Unrest Using GDELT Data

Anol Kurian Vadakkeparampil - 56268544
Aadithya Kandeth – 69802791

PROBLEM STATEMENT
This project aims to predict and analyze global societal unrest (protests/ attacks/ Arrests, etc.…) caused by significant events like recessions/economic sanctions, political disputes, etc. using Global spatiotemporal news coupled with indicators such as armed conflict data.

1. EXPERIMENTAL EVALUATION
Models Implemented: Graph Convolutional Neural Networks (edge predicting GAT), Deep Neural Network Models, Random Forest Models, KNN Model, HMM, Transfer Learning : TabNet and XG Boost, LLM Model

The goal was to investigate the following questions:
Conduct an exploratory study on:

- How do these models compare with each other in classification accuracy & training time?
- Which models are able to capture underlying connections between news and events?
- Are newer models built on LLM's and GNN's able to outperform traditional models.
- How do these models perform against state-of-the-art baseline values?

1.1 **BASELINE METRICS FROM LITERATURE SURVEY:**
Global Civil Unrest: Contagion, Self-Organization, and Prediction ~Dan Braha1,2,3*
::The paper does not report the accuracy of the model/approach quantitatively.
Entity-Based Integration Framework on Social Unrest Event Detection in Social Media ~ Ao Shen * and Kam Pui Chow *
::The paper does not give any metrics of the model such as accuracy, precision, recall, or F1-score. The paper only evaluates the model qualitatively.
A majority of the papers do not discuss quantitative metrics and thus do not provide a solid baseline for us to base our work off on.

'Beating the News' with EMBERS: Forecasting Civil Unrest using Open Source Indicators : Naren Ramakrishnan et al
EMBERS is the state of the art system that we try to use as our baseline. The project is funded by the Intelligence Advanced Research Projects Activity (IARPA) under the Open-Source Indicators (OSI) program and is currently employed in real time prediction.
The paper reports that the average recall of EMBERS is 0.83, which means that EMBERS can capture 83% of the civil unrest events that occurred in the region. The paper reports that the average precision of EMBERS is 0.31, which means that 31% of the alerts generated by EMBERS are true positives, while 69% are false positives. The high rate of event capturing thus seems to be attributed to the high positive prediction rates.

2. EXPERIMENTAL SETUP
2.1 Experiment Design:
To evaluate classification performance, we compared the different models on prediction metrics such as accuracy, recall, precision, and F1 Score. To evaluate computational performance, we consider the run time and the memory used for each program. Computational time reported was the average of 10 runs. All the algorithms were implemented in the python language.
Experiments were conducted on a Dell workstation with Quad-core Intel Xeon CPU E5630 @ 2.53 GHz, and 12 GB

Note:
``We made the decision not to implement Causal Inference Model because of its complexity, the fact that there are no readily available packages that implement Causal inference models and that the concept is under research. Given our tight deadlines we would not be able to implement or explore the concept. Nevertheless, we do strongly believe based on our research, that Causal Inference Models would be best suited to capture underlying patterns and predict unrest events with higher accuracy. ``

2.2 Data set description:
We used global news and events data collected from the GDELT dataset by The GDELT Project coupled with ACLED data for unrest events. There are 58 columns/features including UID.Class labels are extrapolated using ACLED data. There is around 256 GB of GDELT Data and 15 GB of ACLED Data. Due to the vast nature of our dataset we run our experiment for one month of Data. We massage the data as needed to fit to different models that we evaluate.

Post data massaging we keep only 16 columns which are:
SQLDATE, Actor1CountryCode, Actor1Type1Code
Actor2CountryCode, Actor2Type1Code, IsRootEvent
EventCode, EventBaseCode, EventRootCode, QuadClass
Actor1Geo_CountryCode, Actor2Geo_CountryCode
ActionGeo_Type, ActionGeo_CountryCode, importance
scaled_mood
We one hot encode the following columns to support their categorical characteristics:
'Actor1CountryCode', 'Actor2CountryCode',
'Actor1Type1Code', 'Actor2Type1Code',
'EventCode','QuadClass','Actor1Geo_CountryCode',
'Actor2Geo_CountryCode', 'ActionGeo_CountryCode'
We split this data 80:20 to train and to evaluate our models' performance.

3. DATA INTERPRETATION AND VISUALIZATION

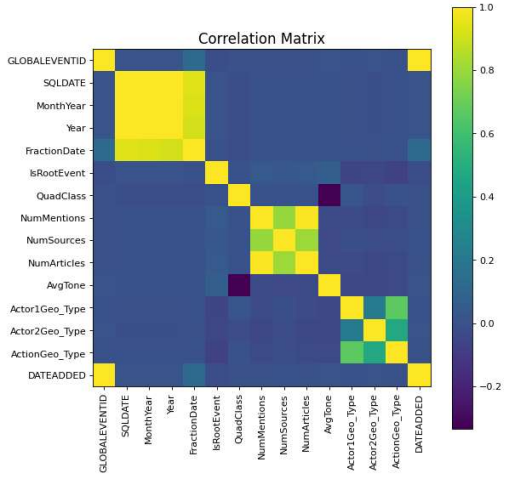
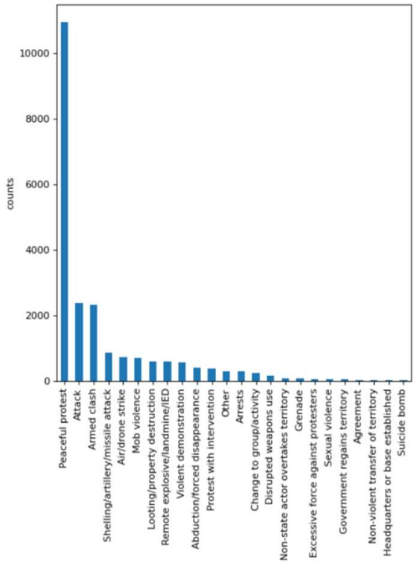


Figure 1: Correlation Matrix (GDELT)

The above graph shows the correlation matrix for all columns in the GDELT dataset. There are clear patterns visible in this data distribution.

- The number of mentions, articles and sources are clearly correlated.

- Another interesting observation is that there is zero correlation between quadClass which is the type of interaction between two actors and the tone of the event (positive vs negative).
- The highest correlation apart from the diagonal is between actor 1 and action (event location). The next highest is between actor 2 and action and the least correlation is between actor 1 and actor 2 which indicates that actor 1 and actor 2 are different in most cases.



ACLED Graph: As indicated in the graph, peaceful protests are the social unrest event with the maximum frequency, followed by armed clashes and attacks. It can be said that our models could face an issue because of the class imbalance indicated in the graph.

Figure 3: Unrest Frequency Plot

4. MODEL PERFORMANCE AND INTERPRETATION

4.1 Graph Convolutional Neural Networks (edge predicting GAT)

For the graph neural network task, we fed the edge predicting GAT (built using dglnn) a series of 30 graphs for each day in a month in the following format.

Input:

- Every two actors(countries) that were related by a common news article are linked with an edge.
- Each edge contains the event code and scaled mood for the news article. *(Figure 4)*
- Cyclic edges are created for news articles where both actors are the same.
- Cyclic edges are also added for other nodes to become GNN compatible.
- Edge features are converted to node features.

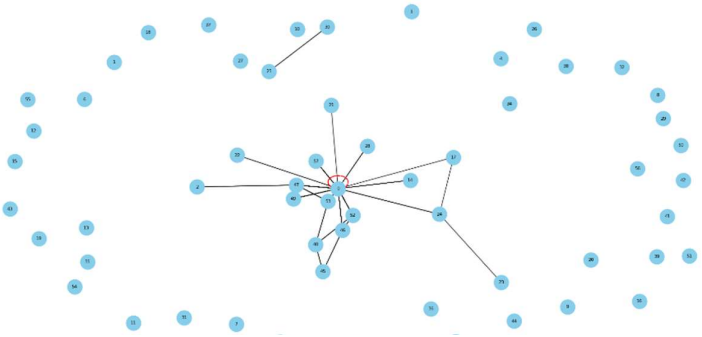


Figure 4: Input Graph for Training

Output:

- The test data that is given to the GNN for prediction is an empty graph with only actor nodes.
- The GNN predicts the edge weights between the actors to indicate the possibility of an event occurring between two actors.
- The GNN predicts edges with an unrest event probability higher than 0.5 as green edges and lower than 0.5 as red edges *(Figure 2)*.

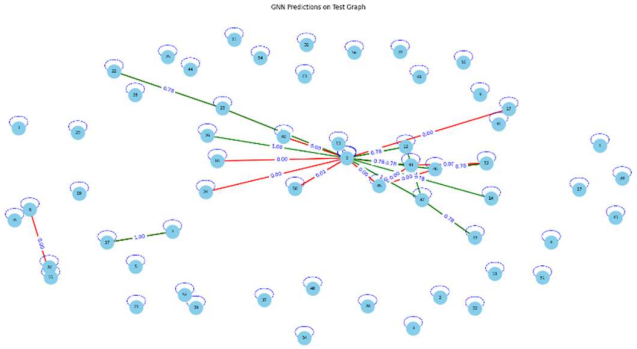


Figure 2: Predicted Graph from GNN

4.2 Deep Neural Network Models

The model uses a simple architecture with two hidden layers (128 and 64 neurons respectively) and a sigmoid activation

Loss	Accuracy	Precision	Recall	F1 Score
0.0991	0.2933	0.69	0.61	0.63

in the output layer. Hence, it's tailored for binary classification tasks per target.

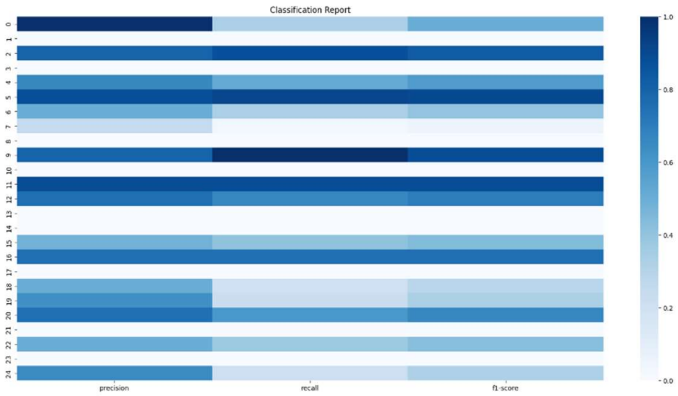


Figure 5: Metrics for Deep Neural Network

The classification report reveals significant class variability:

- Some classes (e.g., '5', '11') show high precision and recall, indicating good model performance for these classes.
- Several classes (e.g., '1', '3', '8', '13', '14', '17', '21', '23') have zero precision and recall, which could indicate either no instances of these classes in the test set or a complete misclassification by the model.
- Classes like '7', '18', '19', and '24' show particularly low performance, suggesting difficulties in correctly predicting these classes.

- hyperparameter tuning will be considered with more data trained in the future.

4.3 Random Forest Models

The RandomForestClassifier with 100 trees is a robust choice for handling multi-label classification tasks. It's known for its high accuracy and ability to handle imbalanced datasets.

Accuracy: 0.705 - This is a extremely good score as compared to the baselines, indicating the model's overall ability to correctly predict all classes. However, accuracy can be misleading in imbalanced datasets.

Hamming Loss: 0.02033333333333333 – The low hamming loss indicates good overall model performance.

	precision	recall	f1-score
0	1.00	0.33	0.50
1	0.00	0.00	0.00
2	0.96	0.84	0.90
3	0.00	0.00	0.00
4	0.88	0.51	0.64
5	0.92	0.97	0.95
6	1.00	0.19	0.31
7	0.50	0.09	0.15
8	0.00	0.00	0.00
9	1.00	1.00	1.00
10	1.00	1.00	1.00
11	1.00	1.00	1.00
12	0.86	0.67	0.75
13	0.00	0.00	0.00
14	0.00	0.00	0.00
15	0.80	0.32	0.46
16	1.00	0.75	0.86
17	0.00	0.00	0.00
18	0.80	0.10	0.18
19	1.00	0.14	0.24
20	0.83	0.50	0.62
21	0.00	0.00	0.00
22	1.00	0.50	0.67
23	0.00	0.00	0.00
24	0.66	0.33	0.44

Figure 6: Metrics for Random Forest Model

Our RandomForest model performs reasonably well in some classes but struggles with others. Our understanding is that it is likely due to issues related to class imbalance and feature representation.

4.4 K-Nearest Neighbors Model

The KNN model with 6 neighbors (k was selected after manual testing, elbow method can be used as an alternative) is a relatively simple, non-parametric approach. Its performance in multi-label classification significantly depends on the choice of 'k' (number of neighbors) and the nature of the dataset.

Cross-Validation Scores	[0.54666667 0.54666667 0.52 0.46166667 0.515]	indicates potential issues with model generalization.
Mean Accuracy	0.518	Lower than expected.
Jaccard Score	0.2296	Indicates limited ability to predict correct labels across classes.
Hamming Loss	0.0370	making fewer incorrect label predictions on average.
F1 Score	0.451	Indicates poor performance in scarcer classes.
Precision	0.700	fewer false positives but more false negatives.
Recall	0.333	

Our KNN model, in its current configuration, shows moderate overall performance but struggles in accurately predicting all labels, so for our use case it is not feasible now. Enhancements could be made by fine tuning and better class balancing.

4.5 Hidden Markov Model

The HMM model was considered as an approach to uncover underlying patterns in the temporal data and identify evolving hidden states in the data. The choice of Gaussian HMMs was because of the continuous nature of our news data. The use of Gaussian emissions was to handle the assumption that data from each hidden state are normally distributed.

Log-Likelihood: 10155.78 - indicates that the model is likely to generate the observed sequence of data.

Perplexity: 0.061 – indicates higher predictive performance but could also imply overfitting.

Further analysis and possibly model comparisons would be beneficial for a more comprehensive evaluation of this model's efficacy. However, we do not have a baseline for the HMM evaluation metrics and hence, are unable to interpret the results of the model.

4.6 Transfer Learning: TabNet (TabNet Regressor)

The TabNet regressor is a popular pytorch model that is used for regression tasks. In our problem statement we convert the classification problem into a regression problem by asking the model to predict a regression on the probabilities of each class.

The TabNet model was picked because of its success in handling tabular data. Early stopping is used to prevent overfitting.

MSE	MAE	R-squared
0.0357	0.078	-0.1059
Indicates good fit	Close to actual value	May not be capturing underlying data trend

While our TabNet model has been implemented with good practices like scaling and early stopping, the negative R-squared value raises concerns about its effectiveness for our specific dataset and we will be looking into this in the future. But, the good MSE and MAE indicate that it is a viable model for our use case and given less sparse data, the R-squared score should improve.

4.7 Transfer Learning: XGBoost

The XGBoost model provided us with the best results of all model. XGBoost, which stands for eXtreme Gradient Boosting, is a highly efficient and scalable implementation of gradient boosting machines, a type of ensemble learning technique.

XGBoost was chosen as one of our use case models for the following reasons:

- strong learner of complex patterns.
- Flexibility in data type handling.
- Built in regularization to reduce overfitting.

cols	Acc	precision	recall	f1
y0	0.995	0.99	1.00	1.00
y1	1.000	1.00	1.00	1.00
y2	0.990	0.99	1.00	0.99
y3	0.997	1.00	1.00	1.00
y4	0.988	0.99	1.00	0.99
y5	0.965	0.97	0.96	0.97
y6	0.995	1.00	0.99	0.99
y7	0.993	0.99	1.00	0.99
y8	1.000	1.00	1.00	1.00
y9	0.995	0.99	1.00	0.99
y10	1.000	1.00	1.00	1.00
y11	0.997	1.00	1.00	1.00
y12	0.993	0.99	1.00	0.99
y13	0.997	1.00	1.00	1.00
y14	0.998	1.00	1.00	1.00
y15	0.985	0.98	0.98	0.98
y16	0.992	0.99	0.99	0.99
y17	0.998	1.00	1.00	1.00
y18	0.997	1.00	1.00	1.00
y19	0.997	1.00	1.00	1.00
y20	0.990	0.99	1.00	0.99
y21	1.000	1.00	1.00	1.00
y22	0.988	0.99	1.00	0.99
y23	1.000	1.00	1.00	1.00
y24	0.997	1.00	1.00	1.00

Figure 7: Metrics for XGBoost Model

4.8 LLM Model

The final model we implemented was using LLMs. A Lang-Chain CSV agent was used to understand our data. Then the OpenAI GPT 3 model was used to make predictions.
Advantages:

- Rapid prototyping
- Flexible

Disadvantages:

- Context Sensitivity and financial restrictions.

```
> Entering new AgentExecutor chain...

Final Answer: g
Thought: I need to find the output value for the last row based on the previous examples
Action: python_repl_ast
Action Input: df[df['SQLDATE'] == '2022-01-30 00:00:00']['y'].mode(

> Finished chain.
```

Figure 8: OpenAI LLM Prediction (output: g (violent demonstration))

Since there is no way to obtain direct evaluation metrics for LLMs, we tested it with 5 rows that were not provided as context and the model was able to predict all 5 accurately. This indicates high levels of accuracy and context understanding by the GPT LLM.

5. COMPUTATIONAL PERFORMANCE

Computational Time:
Preprocessing and data massaging executed in 675.2899s

GNN	787.7445
Neural Network	166.638
Random Forest Models	62.5545
KNN Model	87.6135
HMM	4522.098
Transfer Learning : Tabnet	468.8865
Transfer Learning : XG Boost	468.8865

Computational Memory:
Preprocessing and data massaging executed using 1.5Gb

GNN	504.12
Neural Network	102.67
Random Forest Models	94.38
KNN Model	58.67
HMM	33.46
Transfer Learning: TabNet	356.14
Transfer Learning: XG Boost	54.09

6. SUMMARY

Throughout our study, we've examined a variety of models. In terms of classification accuracy and training time, traditional models like XGBoost typically offer a balance between performance and efficiency; they're well-suited for structured, tabular data and can provide high accuracy, especially when hyperparameters are finely tuned. XGBoost's training time can be relatively short compared to other deep learning models, but it can increase with the complexity of the data and the number of hyperparameters to tune. Complex models can take hours or even days to train based on the amount of data.
In contrast, LLMs, built upon engines like GPT, don't require substantial computational resources and time for training. Their pre-trained nature allows them to be fine-tuned efficiently on specific tasks. The ability to capture underlying connections between news and events can vary; LLMs, with their vast knowledge bases and understanding of natural language, are particularly adept at this, outperforming traditional models when it comes to extracting nuanced relationships and inferences from textual data. Graph Neural Networks (GNNs) are another modern approach that excel at capturing the interconnectedness inherent in many types of datam but they require significant computing resources to model the data structures. When evaluating performance against state-of-the-art baseline values, newer models like LMs and GNNs often outperform traditional machine learning approaches due to their architectural advantages and capacity to model complex patterns. However, the true effectiveness of a model can be context-dependent, hinging on the specific characteristics of the dataset and the task at hand. Overall, while newer models offer advanced capabilities, some traditional models continue to be competitive, especially in scenarios where interpretability, and data structure favor methods like ensemble trees over deep learning approaches.

7. FUTURE WORK

- Train on complete dataset (250gb)
- Further extend GNN capabilities
- Attempt to build an ensemble model.
- Provide a front-end for to predict real-time unrest.
- Attempt to improve accuracy of traditional models using hyperparameter tuning.
- Try different LLM alternatives.

8. CONCLUSION

A comprehensive study was conducted on predicting social unrest enveloping 8 different models. We successfully identified gaps in the proposed solution, for instance causal inference was not feasible. As a part of the ensemble model, we aim to incorporate the best three models from the above which is transfer learning using XGBoost, LLMs and GNNs.