Attempting Predictions in the Israel-Palestine Conflict: An ML and Econometric Approach

Presented by:

Arya Amarnath

Professor - Dr. Emily J. King

DSCI 478: Senior Capstone Project

Colorado State University

May 1, 2024

Abstract:

In this research study, I will analyze the ongoing Israel-Palestine conflict utilizing predictive modeling techniques via data sourced from ACLED, the Armed Conflict Location and Event Data Project. ACLED has categorized political violence into six 'event-types' consisting of 25 violent and non-violent sub-types. (ACLED, 2024) Focusing on predicting the 25 sub-event types, I will research patterns and trends to understand why these specific events and interactions happen under various reproducible conditions. Attempting to replicate time-series forecasting on a standard classification problem by creating temporal features and utilizing econometric techniques. Combining data preprocessing and econometric techniques, including log transformations, lagging, encoding, and feature selection methods utilizing mutual importance and Random Forest scores, this study will utilize Long Short-Term Memory models. Delving into data analysis and visualizations of the models' predictions, the goal is to create a framework that can help support conflict resolution through predictive forecasting.

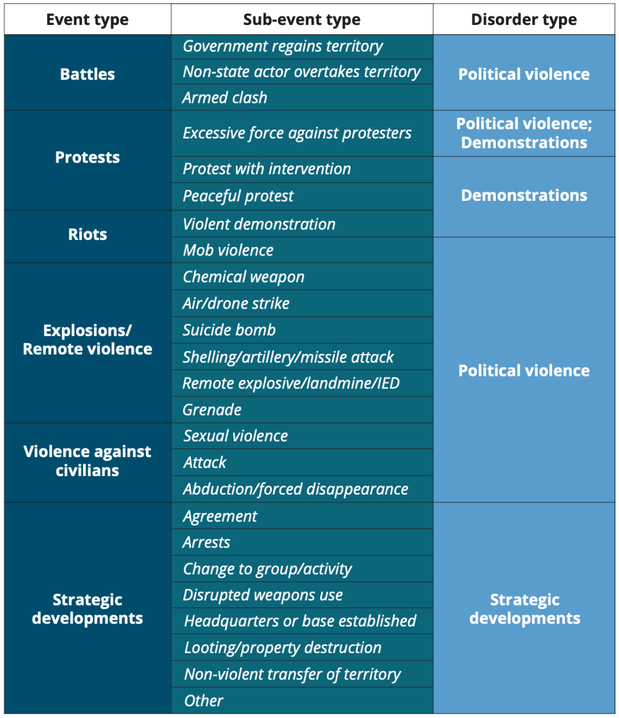
The Palestine-Israel conflict is one of the most enduring and politically charged disputes in modern history, involving deep-rooted national, territorial, and religious tensions. It has significant implications not only for regional stability but also for international peace and security. Understanding the nuances of this conflict through data-driven methods can help contribute to a deeper understanding of its dynamics, potentially leading to effective solutions implemented within this conflict and others around the world. Utilizing data from the Armed Conflict Location & Event Data Project, ACLED, I will be using data-driven methodologies to examine the conflict through the lens of predictive classification. This approach allows for the analysis of political violence and demonstration events with a granularity that traditional analyses may miss. By focusing on attempting to predict the 25 sub-event types defined by ACLED, this study seeks to uncover underlying patterns that may trigger conflict-related events.

The primary objective of this research is to explore how different predictive modeling techniques can enhance our understanding of the dynamics of the Israel-Palestine conflict. Specifically, it aims to understand the various factors and conditions under which types of conflict events are likely to occur and to predict past or future instances of these events. This will involve a combination of machine learning models such as decision trees, logistic regression, recursive neural networks (RNN's), and econometric concepts utilizing time-series analysis tailored to the unique nature of this conflict data.

Research:

Predictive modeling offers a transformative approach to understanding complex systems such as political conflicts. Utilizing historical data, these models can identify patterns and predict future outcomes. The ACLED dataset provides detailed records that include categorical and numerical values including event types, locations, dates, and involved parties from April 2021 to April of 2024. ‘The ACLED Event Type Chart, as shown to the side, breaks down ‘event_type’,‘sub_event_type’, and ‘disorder_type’ providing their respective categories. This dataset was quite large with 38,130 observations and 32 features as of April 19th, in much need of data pre-processing. I chose to predict the sub-event type as opposed to the event type since it was more granular which theoretically allows for more intrinsic relationships to be discovered.

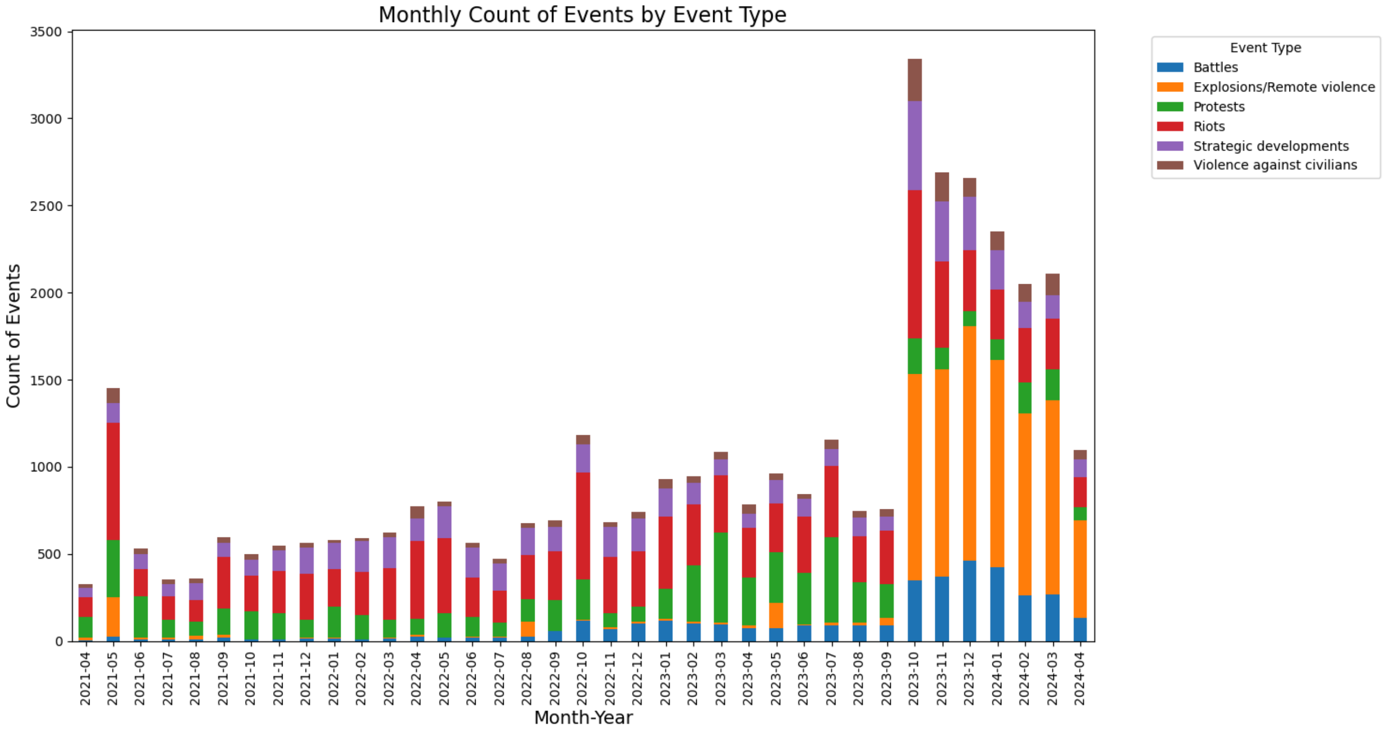
Table 2: ACLED Event Types



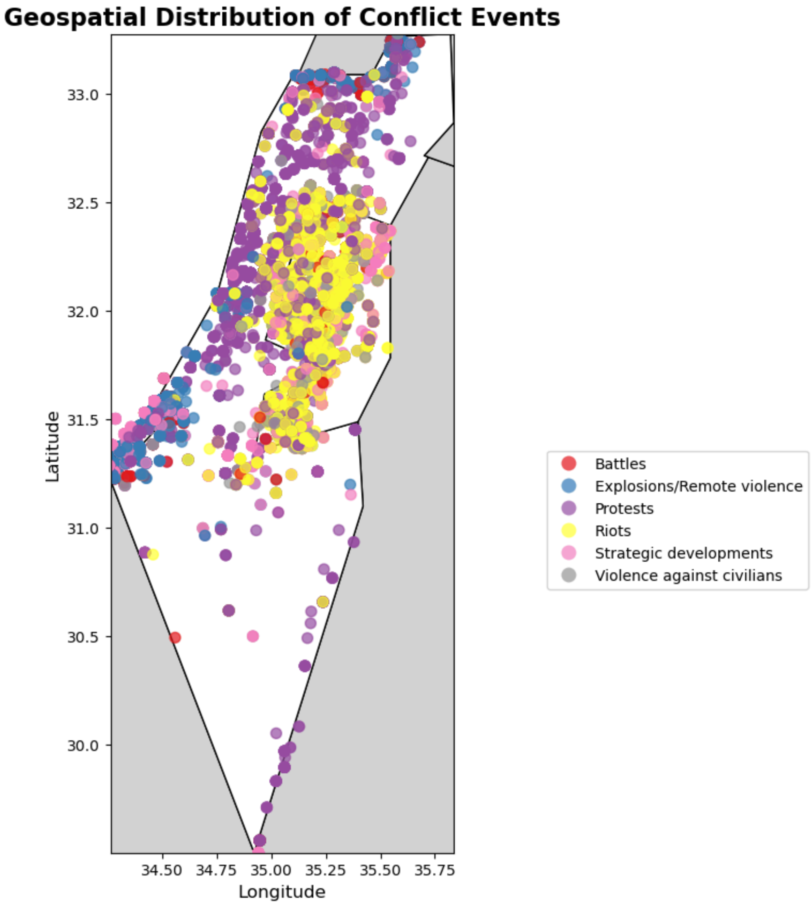
Event type	Sub-event type	Disorder type
Battles	*Government regains territory*	**Political violence**
	Non-state actor overtakes territory	
	Armed clash	
Protests	*Excessive force against protesters*	**Political violence; Demonstrations**
	Protest with intervention	**Demonstrations**
	Peaceful protest	
Riots	*Violent demonstration*	
	Mob violence	**Political violence**
Explosions/ Remote violence	*Chemical weapon*	
	Air/drone strike	
	Suicide bomb	
	Shelling/artillery/missile attack	
	Remote explosive/landmine/IED	
	Grenade	
Violence against civilians	*Sexual violence*	
	Attack	
	Abduction/forced disappearance	
Strategic developments	*Agreement*	**Strategic developments**
	Arrests	
	Change to group/activity	
	Disrupted weapons use	
	Headquarters or base established	
	Looting/property destruction	
	Non-violent transfer of territory	
	Other	

(ACLED Codebook, 2024)

This granular data allows for a comprehensive study of the conflict's dynamics over time, revealing trends and patterns not shown in media reports or regular statistics alone. For instance,



these two visualizations provide insight into ‘event_type’ based on their frequency and geospatial location. Ever since October of 2023, we can see a dramatic spike in the amount of events coinciding with the October 7th date in which Hamas attacked Israel with Israel promptly responding by mobilizing the IDF, Israeli Defense Force, with bombing runs on the Gaza Strip following shortly. (CPA, 2024) This is highlighted by the orange section on the bar plot above and the blue dots within the geospatial data. This geospatial distribution also presents the high number of riots within the West Bank where countless families both Palestinian and Israeli reside. I also used this time to do some basic statistical theory on my data where I found that there were quite a few columns with high variances and correlation and quite a bit of class imbalance within this dataset due to its high dimensionality and wide range of features.



Methodology:

Starting, the data will need to be cleaned and pre-processed in a way that allows for time to be tracked and modeled. After basic data cleaning including removing NA values, taking out duplicates, and re-formatting the data labels for readability, I started preparing for time-series predicting. This required the creation of new temporal features such as day, week, month, the time since the last observation of that column, rolling averages, and a cumulative count for the event type as shown above in the graphs. These features allow for the modeling of seasonality and trends at different granularities which wouldn't have been possible with the given features of the dataset. For example, certain types of conflict events may be more prevalent on specific days of the week or during certain months due to political, religious, or social factors, potentially enhancing the model's ability to detect and learn from these periodic trends. Creating new features such as the time since the last event observation can be particularly insightful as well. This feature, known as "recency," captures the temporal distance between events and is crucial in understanding and predicting the likelihood and timing of subsequent events. Rolling averages, a specific series of averages along the dataset, were used to smooth out short-term fluctuations highlighting longer-term trends or cycles in the time series. The goal was to utilize a rolling average to smooth out daily fluctuations to reveal a consistent pattern of escalation/de-escalation over a specified time window however this ended up failing due to class imbalance issues. (Geeksforgeeks, 2024)

To ensure our time-series data adheres to the prerequisites for analysis, we must attend to the variance and stationarity within our dataset. Stationary time series data, where the mean, variance, and autocorrelation are constant over time, is essential for generating reliable predictions from our models. Non-stationary data, on the other hand, can mislead models due to the presence of underlying trends and cycles that influence the observed values. In the case of the ACLED dataset, non-stationarity could manifest as shifts in the frequency of conflict events over time due to various factors such as policy changes, ceasefire agreements, or escalations in hostility. To detect this, I used the Augmented Dickey-Fuller (ADF) test. This statistical test helps us determine if the data is non-stationary by testing for the presence of a unit root. When the test indicates non-stationarity, differencing is a common method for transformation, removing trends and seasonality from the series and stabilizing the mean. (Grace-Martin, 2019) In conflict data, we often find that the distribution of events can be skewed with sporadic spikes in intensity, such as a sudden outbreak of violence leading to a high casualty rate. To combat this and achieve homoscedasticity aka constant variance, log transformations are applied. The logarithmic function reduces the effect of these spikes, pulling in the long tail of our distribution and creating a more symmetrical, bell-shaped curve. This is imperative for linear time-series models which assume a constant variance in the error terms. (Wang, 2021)

Finally, the concept of lagging variables was introduced to incorporate the temporal ordering of events. This step is vital in conflict analysis due to the autocorrelated nature of conflict data, where past events often influence future ones. By including lagged variables, the past values of the same series, we allow our model to learn from the past in order to predict the future. All relevant features were lagged by a factor of '3' utilizing the autocorrelation function. The data also has to be non-multicollinear meaning that variables cannot be highly correlated to each other otherwise they end up clouding the model's performance as it cannot differentiate between the two variables. In order to check this, I utilized Variance Inflation Factoring (VIF) in order to check how much of the variance of the coefficient is being inflated by said multicollinearity. (Ananth, 2021). That being said, I was unable to properly handle the multicollinearity within the data as the VIF didn’t result in any meaningful results, and utilizing alternate tests such as the Breusch-Pagan test wasn’t applied to their full potential leaving a partially multi-collinear dataset which will impact our predictions but more likely the forecasts.

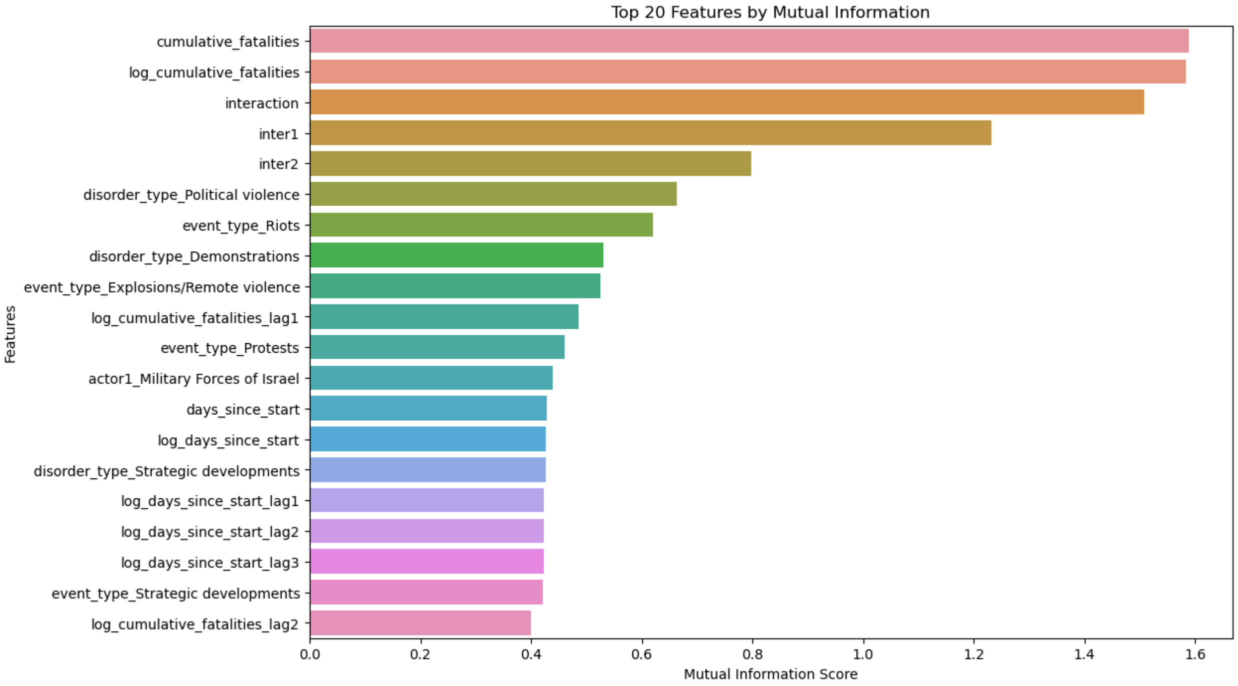
After transforming our data into a format somewhat usable for time-series, the next step to getting our data ready for our model is encoding. Since our data includes categorical variables, they will need to be converted into numerical format so our models can actually input them. I utilized two types of encoding:

- Label Encoding: Assigns a unique integer to each category
- Used for ordered variables: ‘actor1’, ‘actor2’
  - For example within actor1: ‘Military Forces of Israel = 1 if the actor is a part of the ‘Military Forces of Israel’ otherwise they equal 0. Each unique category is transformed into a binary vector, where the presence of an interaction type is marked by '1' and the absence by '0' ensuring that no artificial ordinal relationship is introduced among these interaction types.
- One-Hot Encoding: Creates a binary column for each category
  - Used for ordered categorical variables: ‘event_type’, ‘actor’, ‘day_of_week’
  - For example ‘day_of_week’ becomes ‘Monday = 1’, ‘Tuesday = 2’, ‘Wednesday = 3’. The days of the week are now represented by integers, maintaining a natural order that could be significant for our time-series analysis.

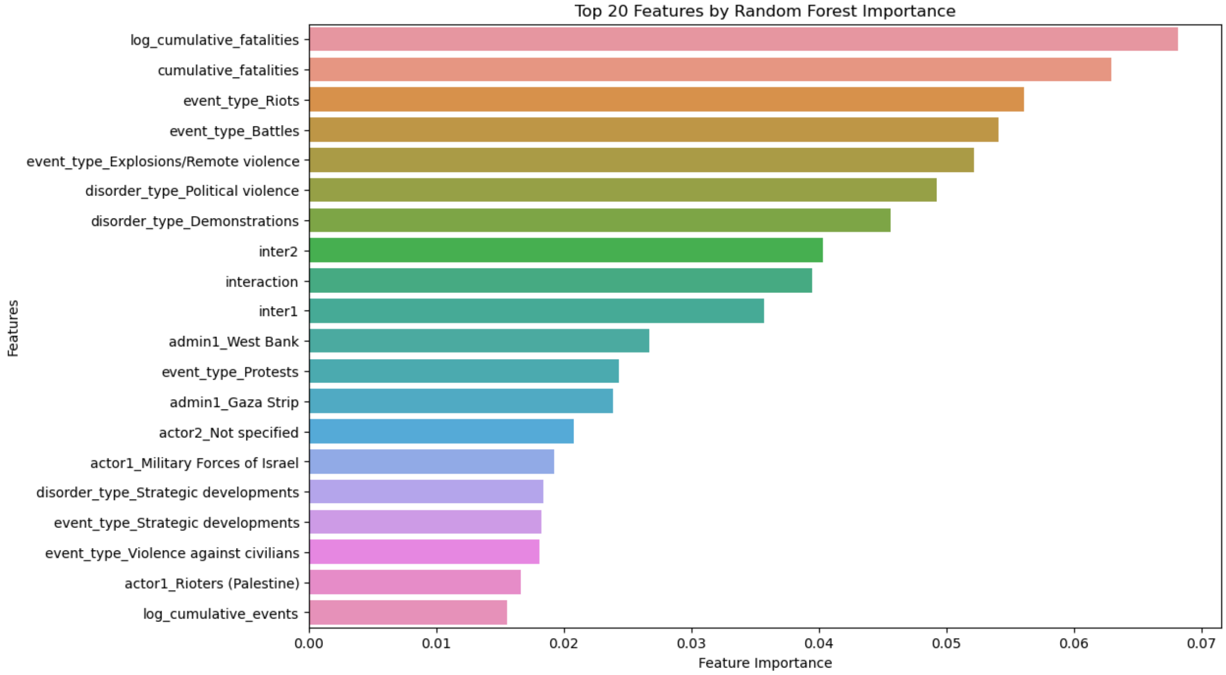
Encoding allows for our model to input the data however it also greatly can increase the dimensions of your dataset especially when using one-hot encoding which will create a new column for each unique observation of that feature. This increases our dataset’s dimensions from (38090 rows × 155 columns) compared to our original size (38130 rows × 32 columns).

Once the dataset is encoded, the next step is to identify which features are most relevant to predicting our target variable, the 'sub_event_type'. Feature selection helps in reducing dimensionality, improving model performance, and decreasing computational cost which in this case is extremely important given our huge dataset (38090 x 155). To do this, I utilized mutual importance and a Random Forest classifier together.

- Mutual Importance: Measures the dependency between two variables, in this case, between each feature and the target variable (‘sub_event_type’). The higher the value, the stronger the relationship between the feature and the target.



- Random Forest Scores: Random Forest is an ensemble learning method known for its feature importance scores, which are computed as the average decrease in impurity across all trees in the forest when splitting on a feature.



The two plots above show the top 20 features of both mutual importance and Random Forest feature importance, where similar features such as 'log_cumulative_fatalities' can be observed at the top of both. This approach aims to mitigate the risks of overfitting by excluding less critical features that may introduce noise rather than predictive value. In order to create my final dataset, I ended up taking the 80th percentile (top 20%) of both groups resulting in reducing the number of observed features from 155 to 24 allowing for a harmonized, robust set of predictors creating a final dataset size of (30472 rows × 24 columns).

Models:

Splitting my dataset into a train/test split of 80% training and 20% test, I initially utilized baseline models such as logistic regression and decision trees to establish a benchmark for performance. These models useful for their simplicity and transparency provided an initial understanding of the relationships within the data. Logistic regression offered insights into the linear relationships (if any) of the classes within the 'sub_event_type' however with such a weak relationship between variables after the statistical analysis at the beginning, this was expected. Decision trees on the other hand provided a more nuanced view of the non-linear interactions between features and at the least resulted in a higher accuracy than the logistic regression. These baseline models had results that left little interpretation and their purpose was simply to test benchmarks. To utilize the temporal dynamics present in the data, Long Short-Term Memory (LSTM) networks were introduced. LSTMs are a type of recurrent neural network (RNN) that is well-suited for time-series data as they can capture long-term dependencies and patterns across time which are crucial in the prediction of conflict events that are inherently sequential and time-bound. Regular RNN's often struggle with the vanishing gradient problem where models struggle with learning from sequential data and LSTM solves that problem by selectively "holding" and "forgetting" information through a gating mechanism. This gating mechanism allows the flow of information going in and out of a LSTM's memory cell, for the storage of large time-based data, allowing them to capture long-range dependencies more effectively.(Ananth, 2021) Three types of LSTMs were tested each with unique hyperparameters: LSTM with no feature selection or embedded layer (control), feature selection LSTM, and feature selection LSTM with an embedding layer.

These models had two actual *LSTM* layers each with 50 units and were run through *Dropout layers* to help reduce overfitting. The final *LSTM* had an *Embedded* layer that transforms features into dense vectors of fixed size, capturing the similarity and relationships between variables in a lower-dimensional space. The *Embedded* layer allowed for the adjustment of the '*Time-Steps'* parameter which had to be set to 24 to match our number of features however the embedded layer which allows for lower-dimensional calculations can lower or reduce that number as needed! This *Embedded* layer was set to run through every unique category which then gets passed to the final *Output* layer. This *Output* layer was compiled using the *'Adam'* optimizer, an application of stochastic gradient descent. The models were fit on 50 epochs, a number selected after multiple attempts of iterating through 5, 10, 20, 50, and 100 epochs. I also tested between batch sizes 32 and 64 with a split of 10% for validation. The loss function selected was '*Categorical Crossentropy'*, a common multi-classification function built by taking the negative log-likelihood of the true class labels given the predicted class probabilities. This allows the model to penalize itself heavily when it's confidently wrong and less when it's close to the correct answer. (Mehmood, 2022) To select the optimal parameters for my LSTMs, I used a grid search to run through specific values for these parameters in my tuning: *Time-Steps, Number of Features per Time-Step, Number of Neurons, Dropout Rate, and the Output Class.* This grid search iteratively trains models based on specific hyperparameter configurations that I would input into my model. (Brownlee, 2020) The *Number of Features per Time-Step* and *Output Class* both had constant values of 24 (# of features) and 20 (unique values of target variable). After multiple training rounds and iterations of removing, checking, and testing different values, I ended up finalizing these values:

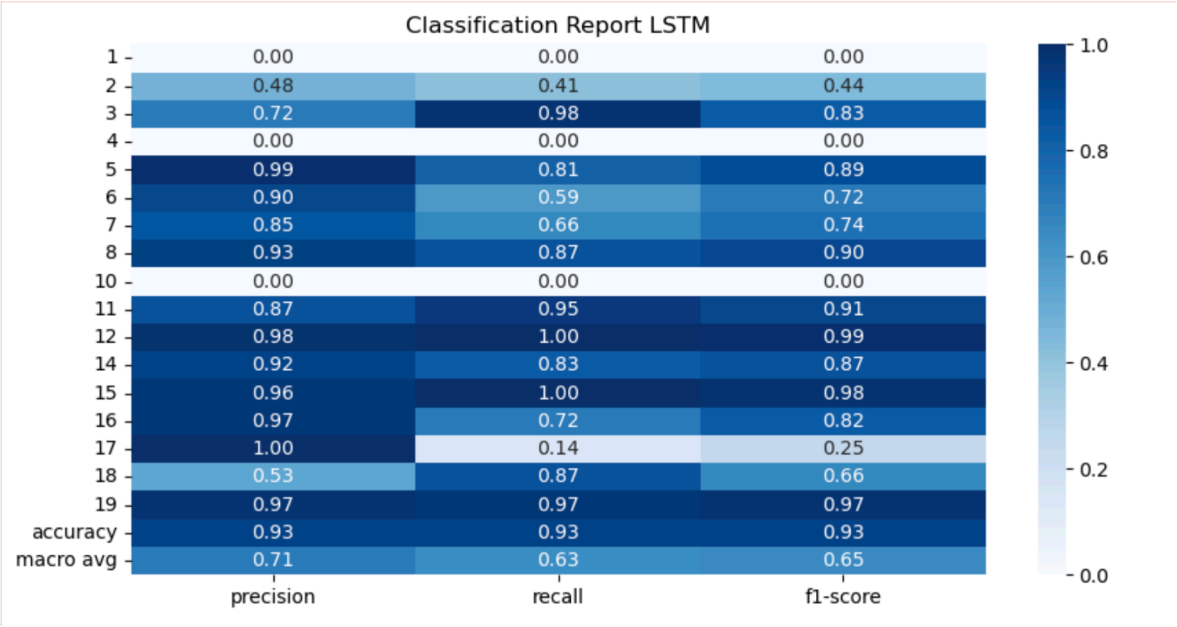
	Time Step	# of Features	# of Neurons	Dropout Rate	Output Class
Control LSTM	24	24	50	0.2	20
LSTM No Embedded Layer	24	24	50	0.1	20
LSTM w/ Embedding Layer	14	24	100	0.2	20

The time-step was experimented with on 7, 14, 24, and 30 days as they represent every period that the model will calculate. The dropout rate was tested between 0.1-0.5 where the ideal was met around 0.2 or 0.1 depending on the time-step.

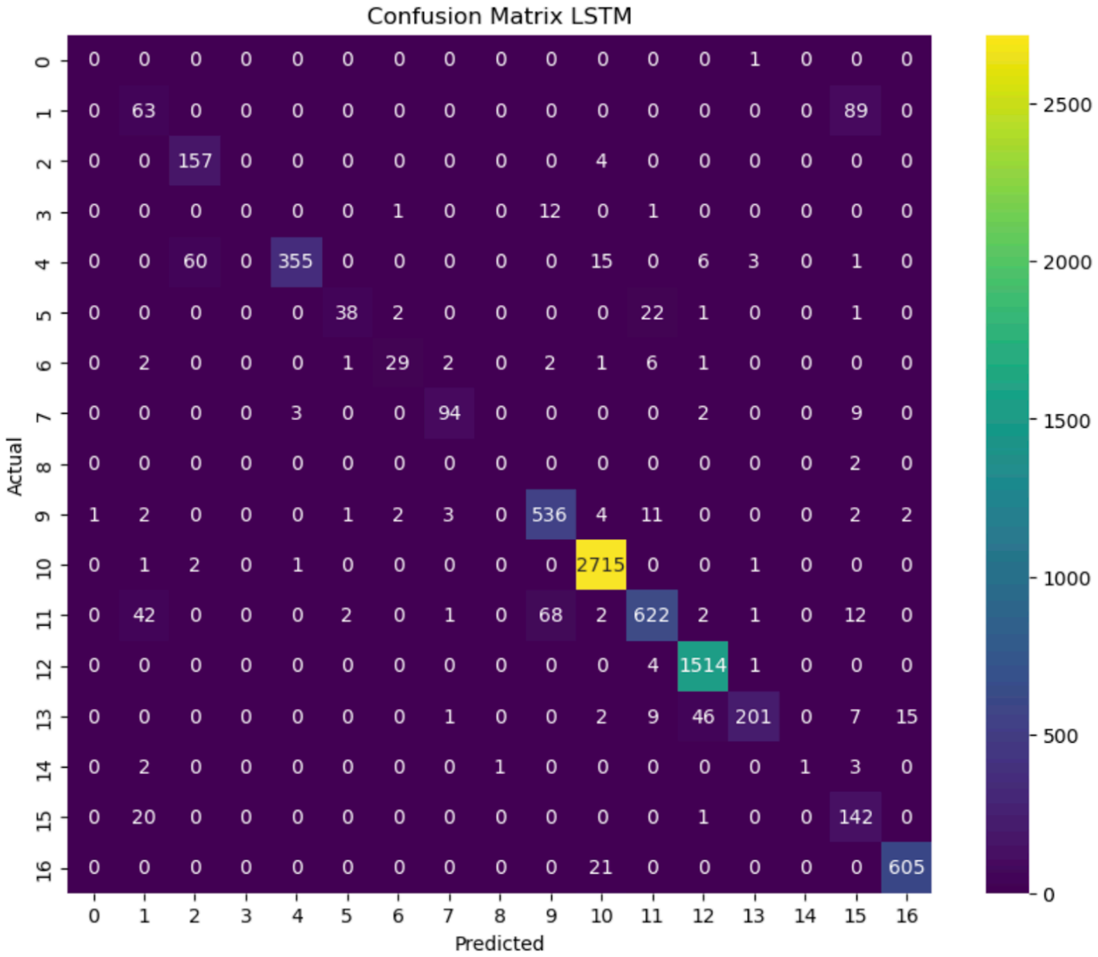
Results:

Analyzing our results from all of our models, specifically the LSTM and VAR models which include predictions against our test data, we can see promising results within our more complicated models as shown below. The LSTM model with an embedding layer achieved the highest accuracy of 84%, highlighting its ability to contextualize and differentiate between the various types of events within the conflict's dataset. The ability of the embedding layer to capture the essence of categorical variables proved beneficial. Contrastingly, the LSTM model without an embedding layer demonstrated a slightly lower accuracy of 79%. The VAR model had slightly worse results with the accuracy being 89% and the lowest error rate. Its slightly lower performance could be attributed to its linear nature which might be less capable of capturing the non-linear dependencies within our data.

	Logistic Regression	Random Forest	LSTM No Feature Selection	LSTM No Embedded Layer	LSTM Embedded Layer
Test Accuracy	0.94	0.66	0.31	0.515	0.84
Test MAE	-	-	8.31	3.94	0.51
Test MSE	-	-	50.39	39.69	5.83
Test RMSE	-	-	9.15	6.30	2.42

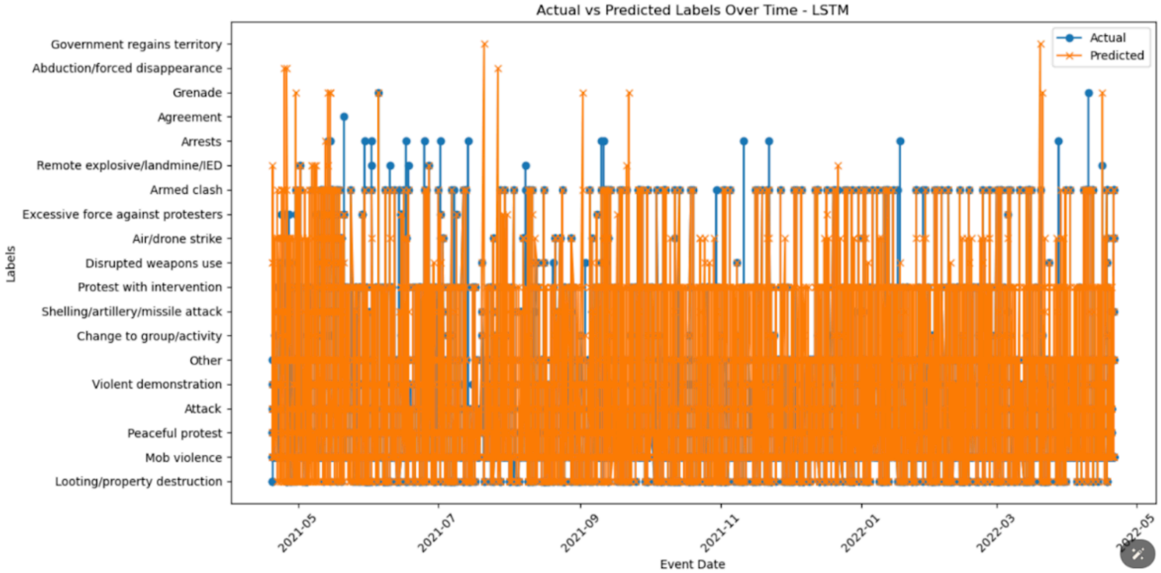


This classification report helps highlight the black box that is our model and goes into how it performed across our different sub_event_types (y-axis). Sub-events in a darker blue such as 'Peaceful protest' (label 5) and 'Attack' (label 12) exhibit high precision and recall showing the model's strong predictive capability for these events. On the other hand, the lighter sub-events like 'Government regains territory' (label 16) and 'Abduction/forced disappearance' (label 17) have notably low scores indicating difficulty in capturing the patterns of these events strongly due to imbalances within the data. The white colored rows show sub-events like 'Other' (label 0) and 'Violent demonstration' (label 1) have zero precision and recall showing that the model fails to understand these.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	89	0
2	0	0	157	0	0	0	0	0	0	0	4	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	12	0	1	0	0	0	0	0
4	0	0	60	0	355	0	0	0	0	0	15	0	6	3	0	1	0
5	0	0	0	0	0	38	2	0	0	0	0	22	1	0	0	1	0
6	0	2	0	0	0	1	29	2	0	2	1	6	1	0	0	0	0
7	0	0	0	0	3	0	0	94	0	0	0	0	2	0	0	9	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
9	1	2	0	0	0	1	2	3	0	536	4	11	0	0	0	2	2
10	0	1	2	0	1	0	0	0	0	0	2715	0	0	1	0	0	0
11	0	42	0	0	0	2	0	1	0	68	2	622	2	1	0	12	0
12	0	0	0	0	0	0	0	0	0	0	0	4	1514	1	0	0	0
13	0	0	0	0	0	0	0	1	0	0	2	9	46	201	0	7	15
14	0	2	0	0	0	0	0	0	1	0	0	0	0	0	1	3	0
15	0	20	0	0	0	0	0	0	0	0	0	0	1	0	0	142	0
16	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	605

This confusion matrix presents the model's predictions against the actual labels. The main diagonal represents correct predictions, with the color indicating the count of correct predictions. High correct predictions are observed for 'Peaceful protest' (label 5), 'Attack' (label 12), and 'Armed clash' (label 13) indicating the model's strong suit in these areas. Off-diagonal elements show where the model confuses one event type with another, for example, 'Peaceful protest' (label 5) is sometimes misclassified as 'Protest with intervention' (label 14).



To truly visualize our predicted values, I created an Actual vs Predicted plot as shown above. The proximity of the predicted (orange) to the actual (blue) points indicates how well the LSTM model tracks real-world events. Discrepancies between actual and predicted points for other event types indicate where the model's predictions diverge. The chart illustrates the model’s temporal predictive power, indicating how well the LSTM captures the timing and frequency of events.

One of the main challenges encountered in this research was data imbalance because certain sub-event types were significantly more prevalent than others leading to skewed distributions. This can potentially bias the predictive model by favoring the majority class. Addressing this imbalance required techniques such as oversampling minority classes and taking the difference of values to attempt a fair representation of all event types in the training data. Another serious issue that I found was dealing with heteroskedasticity and multi-collinearity within my data, I had originally planned to utilize more time-series models such as a VAR or

VECM but during my implementations, I struggled so much with attempting to reduce the multi-collinearity between my data, that I was never able to successfully implement them into my code. This also impacted forecasting into the future as my data just simply wasn't colinear which resulted in extremely low accuracy predictions with an extremely high variance. Looking forward, to improve our models, future research could focus on incorporating additional temporal variables and ensuring that the data is useable within a time-series format. I quickly found out that my project would turn from forecasting into classification prediction simply because I chose a categorical dataset with very few temporal features which is extremely necessary in a time-series forecast. Even by creating my variables, I wasn't able to accurately forecast the future but instead was able to achieve an adequate 92.8% prediction rate on historical values. This allows the model to better capture the interactions between factors driving conflict dynamics. Another would be utilizing natural language processing (NLP) techniques to extract information from textual data, such as the 'notes' and 'source' columns in the dataset. By analyzing the text for sentiment, context, or relevant keywords, new valuable insights appear that could enhance the predictive accuracy of the models. All of this could be implemented into an early-warning system for conflict monitoring and prevention. By continuously analyzing real-time data streams and integrating machine learning/NLP, this system could provide timely alerts and actionable insights to governments and humanitarian organizations. This could play a crucial role in mitigating the impact of conflicts and fostering stability in volatile regions.

In conclusion, this research project aimed to leverage predictive modeling techniques to better understand and potentially mitigate the Israel-Palestine conflict. By attempting econometric and machine learning methodologies within the ACLED dataset, we have gained valuable insights into the patterns and dynamics of conflict events mainly within a prediction classification sense. I was unsuccessful in the task of implementing a forecast that achieved shareable results. Among the models explored, the LSTM with an embedding layer emerged as the most promising, achieving an accuracy of 84%. This underscores the importance of contextualizing and differentiating between the various types of events within the conflict dataset. Even though the model had an extremely imbalanced dataset with weak predictive features, it was still somewhat successful in predicting past values. This entails that with enough tweaking and proper addressing of the multi-collinearity and correlation issues, this dataset can be implemented within a time-series context to achieve forecasted data. Overall, this study demonstrates the potential of data-driven approaches to provide insights into not just the Palestine-Israel conflict but conflicts around the world, attempting to allow better-informed decision-making and intervention strategies.

References:

ACLED. “Armed Conflict Location & Event Data Project Codebook.” *ACLED Data*, 2023, acleddata.com/acleddatanew/wp-content/uploads/dlm_uploads/2023/06/ACLED_Codebook_2023.pdf.

Ananth, Anagha D, and Sujatha Arun Kokatnoor. “An enhanced ensemble of long short-term memory and vector autoregression for energy consumption forecasting.” *Indian Journal of Science and Technology*, vol. 14, no. 43, 12 Nov. 2021, pp. 3227–3236, https://doi.org/10.17485/ijst/v14i43.655.

Bazzi, Samuel, et al. *The Promise and Pitfalls of Conflict Prediction: Evidence from Colombia and Indonesia*, Feb. 2019.

Brownlee, Jason. “How to Grid Search Deep Learning Models for Time Series Forecasting.” *MachineLearningMastery.Com*, 27 Aug. 2020, machinelearningmastery.com/how-to-grid-search-deep-learning-models-for-time-series-forecasting/.

CPA. “Israeli-Palestinian Conflict | Global Conflict Tracker.” *Council on Foreign Relations*, Center for Preventive Action, www.cfr.org/global-conflict-tracker/conflict/israeli-palestinian-conflict#:~:text=The%20Israeli%2DPalestinian%20conflict%20dates,into%20Arab%20and%20Jewish%20states. Accessed 1 May 2024.

Grace-Martin, Karen. “Eight Ways to Detect Multicollinearity.” *The Analysis Factor*, 31 Dec. 2023, www.theanalysisfactor.com/eight-ways-to-detect-multicollinearity/.

“How to Calculate Moving Averages in Python?” *GeeksforGeeks*, GeeksforGeeks, 7 Dec. 2023, www.geeksforgeeks.org/how-to-calculate-moving-averages-in-python/.

Mehmood, Yousuf. “6 Tips to Tweak Your LSTM/BILSTM.” *Medium*, Medium, 4 Feb. 2022, medium.com/@yousufdata/6-tips-to-tweak-your-lstm-bilstm-15fd02685c8.

Parmigiani, Giovanna, et al. “The impact of machine learning in predicting risk of violence: A systematic review.” *Frontiers in Psychiatry*, vol. 13, 1 Dec. 2022, https://doi.org/10.3389/fpsyt.2022.1015914.

Wang, Hongxia, et al. “Multiscale convolutional recurrent neural network for residential building electricity consumption prediction.” *Journal of Intelligent &amp; Fuzzy Systems*, vol. 43, no. 3, 21 July 2022, pp. 3479–3491, https://doi.org/10.3233/jifs-213176.