Forecasting 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 be analyzing the ongoing Israel-Palestine conflict utilizing predictive modeling techniques via data sourced from ACLED; the Armed Conflict Location & Event Data Project. ACLED has categorized political violence into 6 ‘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 better understand why these specific events and interactions happen under various reproducible conditions. Combining both data preprocessing and econometric techniques, including log transformations, lagging, encoding, and feature selection methods utilizing mutual importance and Random Forest scores, this study utilizes various prediction 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 the use of predictive forecasting.

Early draft, need to cite + finish MAE / RMSE code and add it in.

The Palestine-Israel conflict is one of the most enduring and politically charged disputes in modern history, involving deep-seated 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 towards a more nuanced understanding of its dynamics, potentially leading to effective solutions. With a history of the conflict dating back to 1948 (Citation), this deep rooted conflict between these two Middle Eastern countries has been one of the worst tragedies to occur in this generation. Utilizing data from the Armed Conflict Location & Event Data Project, ACLED, I will be using data-driven methodologies to examine the conflict via through the lens of predictive modelling. This approach allows for the analysis of political violence and demonstration events with a granularity that traditional analyses might miss. By focusing on predicting the 25 sub-event types defined by ACLED, this study seeks to uncover underlying patterns and triggers of 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 determine the conditions under which various types of conflict events are likely to occur and to predict future instances of these events. This involves a combination of machine learning models such as Decision Trees, Linear Regression, and Neural Networks and econometric models like time-series analysis, tailored to the unique nature of 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 2021 to 2024. ACLED defines ‘event_type’ as their “fundamental unit of observation”. The ACLED Event Type Chart, as shown to the side, breaks down ‘event_type’, ‘sub_event_type’ and ‘disorder_type’ providing a clear indication of the dataset.

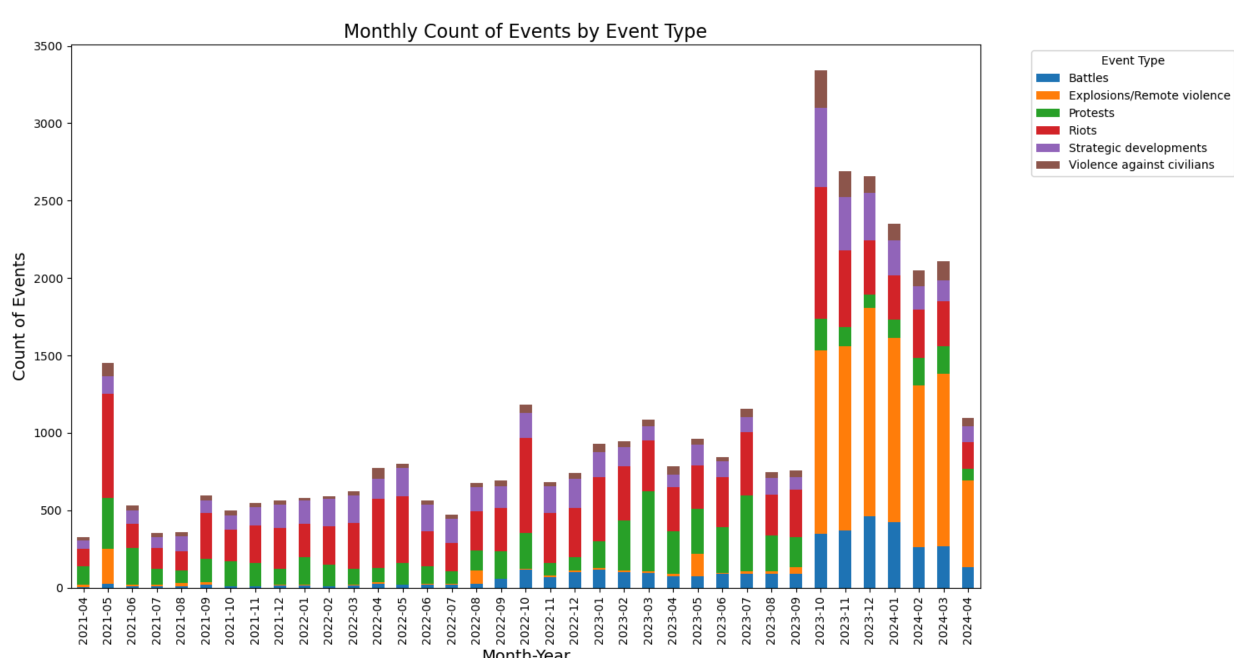
Add sections of research papers talking about Palestine Israel Conflict History, Add Fatality and Civilian Casuality numbers

Table 2: ACLED Event Types

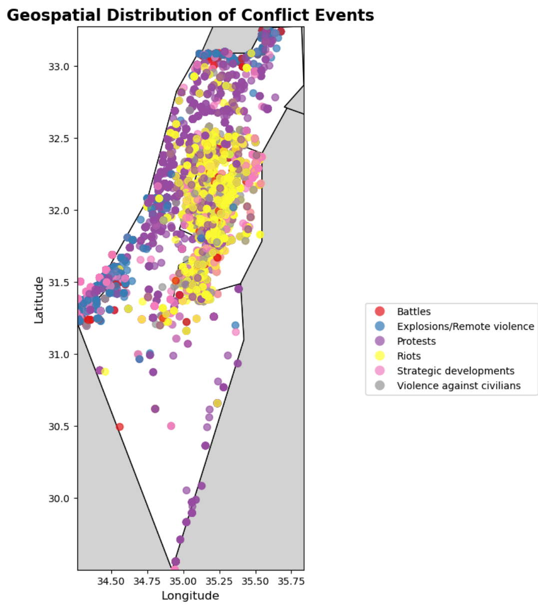
Event type	Sub-event type	Disorder type
Battles	*Government regains territory*	**Political violence**
	Non-state actor overtakes territory	
	Armed clash	
Protests	*Excessive force against protesters*	**Political violence; Demonstrations**
	Protest with intervention	**Demonstrations**
	Peaceful protest	
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 pattern not shown from media reports or regular statistics alone. For instance, these two visualizations provide insight on the ‘event_type’ based on their frequency



and geospatial location. The first shows a bar plot depicting the monthly count of events by event type while the second plot focuses on the geospatial analysis of the conflict events.



Methodology:

Starting off, 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, duplicates and re-formatting the data labels to be viewed easier, I started preparing for time-series predicting. This requires 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. Temporal features such as day, week, and month indicators allow for the modeling of seasonality and trends at different granularities. 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, thus enhancing the model's ability to detect and learn from such periodic tendencies. 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 can be crucial in understanding and predicting the likelihood and timing of subsequent events. Rolling averages, a specific series of averages along the dataset, are used to smooth out short-term fluctuations and highlight longer-term trends or cycles in the time series. For conflict data, a rolling average might smooth out daily fluctuations in violence to reveal a more consistent pattern of escalation or de-escalation over a specified time window. (ADD CITATION)

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. (Citation Here) 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. 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.

After transforming our data, 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 categorical variables: 'event_type', 'actor', 'day_of_week'

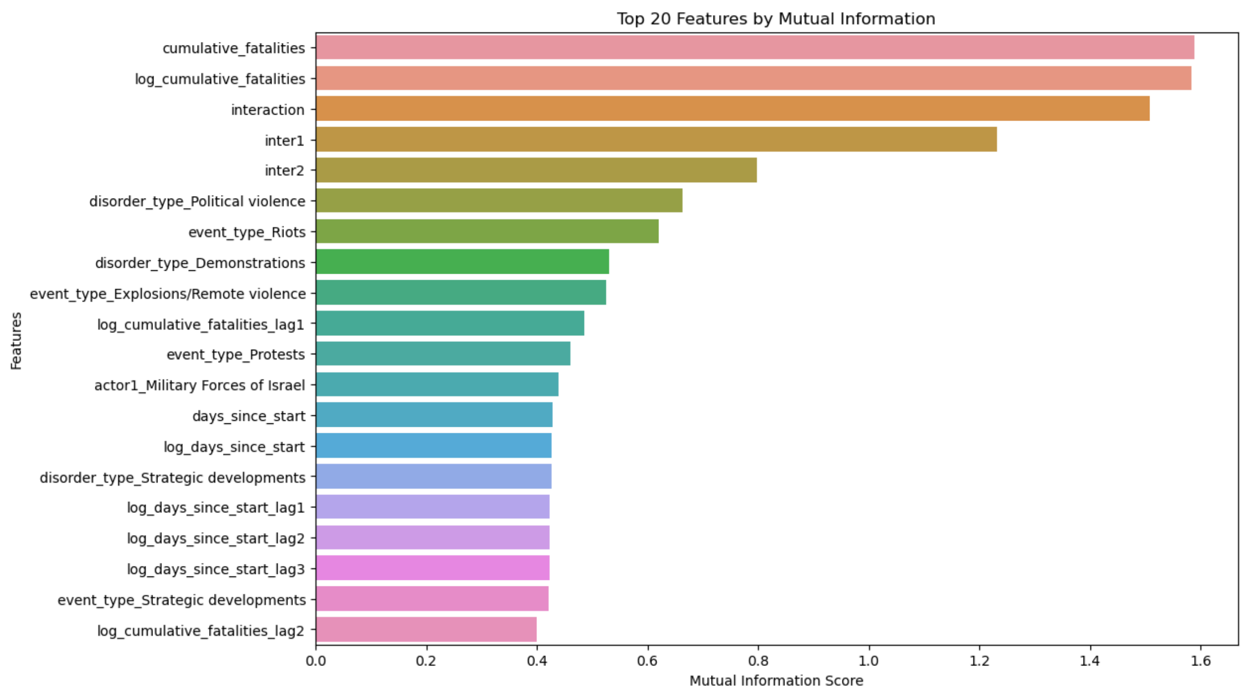
For example 'day_of_week' becomes 'Monday = 1', 'Tuesday = 2', 'Wednesday = 3'. The days of the week are now represented by integers, maintaining a sequence which could be significant for our time-series analysis.

- One-Hot Encoding: Creates a binary column for each category
    - Used for unordered variables: 'interaction', 'inter1', 'inter2'

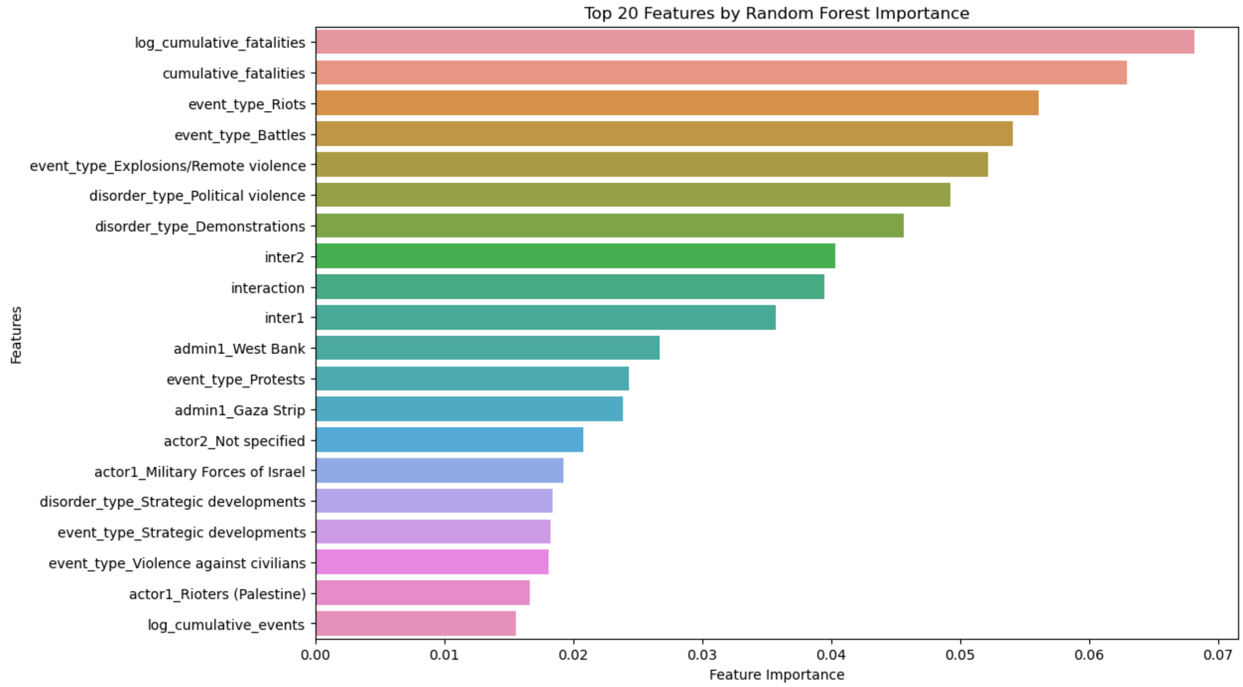
For example within interaction, 'Military engagement = [1, 0, 0]', 'Protest = [0, 1, 0]' ,'Strategic development = [0, 0, 1]'. Each unique category is transformed into a binary vector, where the presence of an interaction type is marked by '1' and the absence by '0' ensuring that no artificial ordinal relationship is introduced among these interaction types. Encoding allows for our model to actually 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. In order 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 is 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 70th percentile (top 30%) of both groups resulting in reducing the number of observed features from 155 to 43 allowing for a harmonized, robust set of predictors.

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'. Decision trees on the other hand provided a more nuanced view of the non-linear interactions between features while being at a risk of overfitting. These baseline models had results that left little interpretation and their purpose was simply to test beginner benchmarks. To harness the temporal dynamics present in the data, Long Short-Term Memory (LSTM) networks were introduced. LSTMs are a type of convolutional neural network that are 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. (Citation Here) Two LSTM architectures were tested:

- LSTM with an embedding layer: This model was designed to process sequences of encoded categorical data, particularly useful for handling the high-cardinality categorical features like 'actor1' and 'actor2'. The embedding layer transforms these categories into dense vectors of fixed size, capturing the similarity and relationships between actors in a lower-dimensional space. This embedding layer was set to run through every unique category.

```
#  LSTM model

model_LSTM = Sequential()
model_LSTM.add(Embedding(input_dim=len(np.unique(y)), output_dim=100, input_length=X_train.shape[1]))
model_LSTM.add(LSTM(50, return_sequences=True))
model_LSTM.add(Dropout(0.2))
model_LSTM.add(LSTM(50))
model_LSTM.add(Dropout(0.2))
model_LSTM.add(Dense(y_train_encoded.shape[1], activation='softmax'))

model_LSTM.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])


model_LSTM.fit(X_train, y_train_encoded, epochs=50, batch_size=32, validation_split=0.1)
```

- LSTM without an embedding layer: By forgoing the embedding layer, this model directly worked with the preprocessed numerical features, relying on the LSTM cells to learn the temporal relationships. This approach is straightforward but may overlook the nuanced patterns that the embeddings can capture.

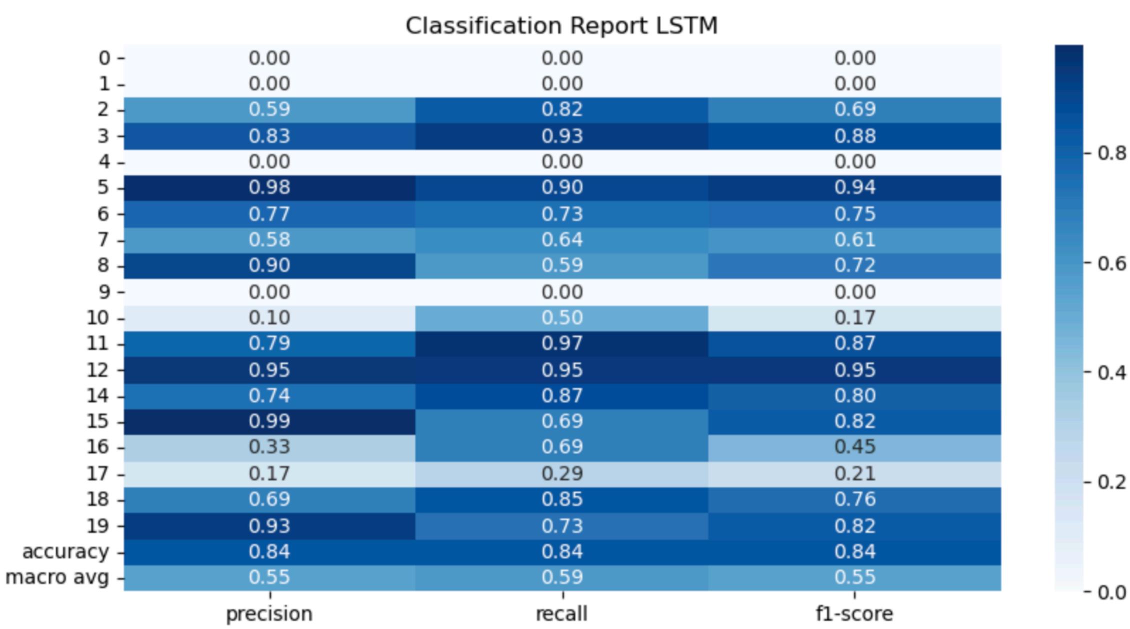
Both models had two actual LSTM layers each with 50 units and were ran through Dropout layers between the actual LSTM layers in order to help reduce overfitting. This goes through a final output player before being complied using the 'Adam' optimizer, an application of stochastic gradient descent. The models were fit on 50 epochs, a number selected after multiple attempts of iterating through 5-100 epochs and had a batch size of 32 with a split of 10% for validation. Following the exploration of LSTM models, next is the Vector Autoregression Model (VAR). VAR allows for forecasting structured time-series data and capturing the linear interdependencies among multiple time-series. VAR models the next time step of each variable as a linear combination of past observations of all variables in the system, making no distinction between endogenous and exogenous variables. Endogenous variables are those that are determined by the model, while exogenous variables are those that are determined outside the model. (Citation Here) VAR is appropriate for data that are stationary or have been transformed to become stationary making it a perfect fit for our data.

```
X_train, X_test = train_test_split(filtered_data, test_size=0.2, random_state=42, shuffle=False)
model_var = VAR(X_train)
fitted_model_var = model_var.fit(maxlags=15, ic='aic')
var_forecasts = fitted_model_var.forecast(X_train.values[-fitted_model_var.k_ar:], steps=5)
print(var_forecasts)
```

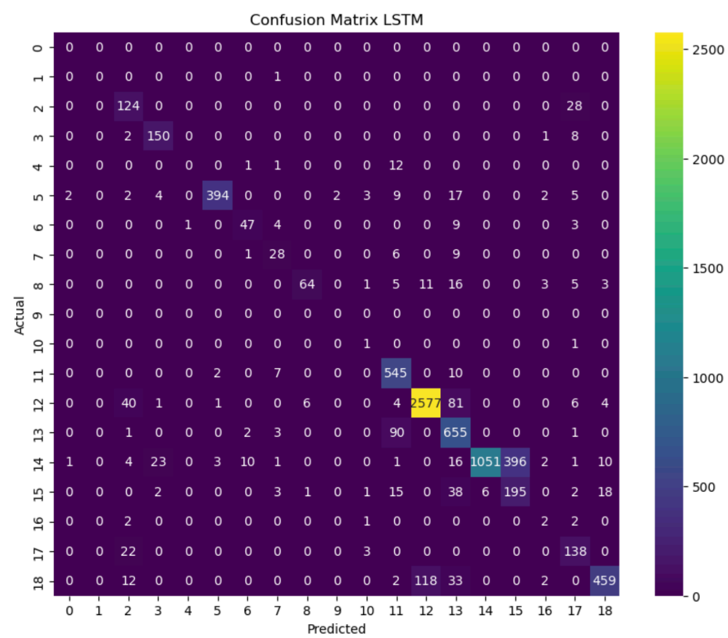
Results:

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 in capturing the non-linear dependencies within our data.

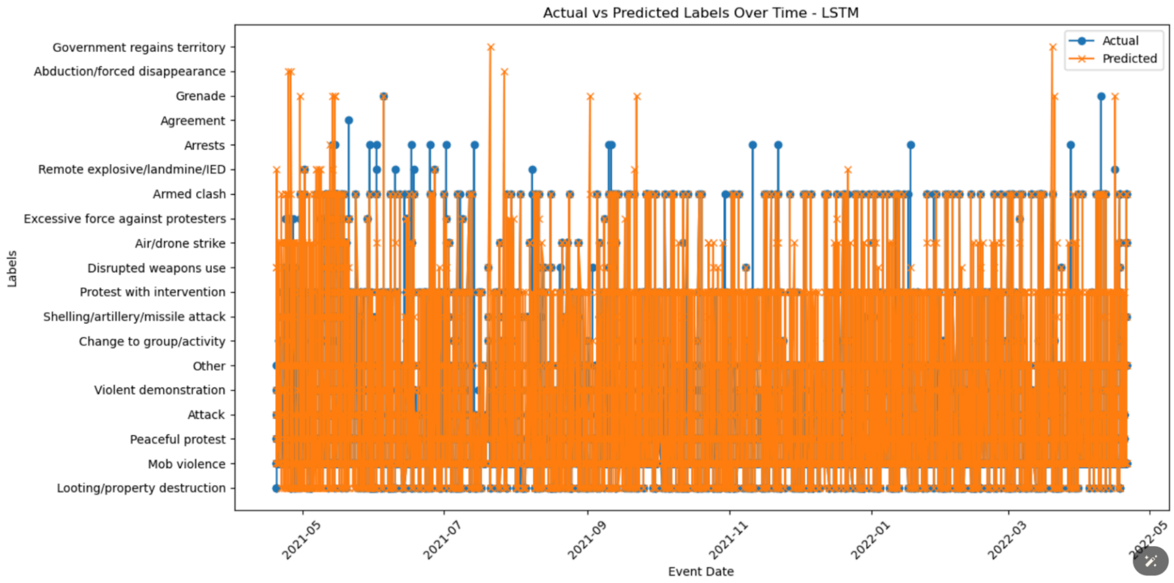
	Lin Reg	Tree	LSTM	LSTM - noembed	VAR
Test Accuracy	0.94	0.66	0.84	0.79	0.82
Test Errors	-	-	15.6%	21.2%	18.7%
Test MAE					
Test RMSE					



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.



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 a 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 with 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 key 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 biasing 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. Looking forward, to improve our models, future research could focus on incorporating additional temporal variables. These could include factors such as historical political events, socio-economic indicators, or even weather patterns, which may influence the occurrence and dynamics of conflict events. 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 which 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.

Conclusion:

In conclusion, this research project aims to leverage predictive modeling techniques to better understand and potentially mitigate the Israel-Palestine conflict. By utilizing econometric and machine learning methodologies within the ACLED dataset, we have gained valuable insights into the patterns and dynamics of conflict events. 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. However, challenges such as data imbalance remain significant hurdles that need to be addressed to improve predictive power. Overall, this study demonstrates the potential of data-driven approaches to provide insights into the Palestine-Israel conflict, attempting to allow better informed decision-making and intervention strategies.

Citations:

ACLED_Codebook_2023.pdf (acleddata.com)

ACLED Data, 2021-2024

ViolencePrediction_Short.pdf (bu.edu)

The impact of machine learning in predicting risk of violence: A systematic review - PMC (nih.gov)

https://cordis.europa.eu/article/id/443344-using-machine-learning-to-identify-political-violence-and-anticipate-conflict