

ElectionShield Live: A Real-Time AI Framework for Detecting Coordinated Narrative Manipulation During Elections

[Daivik Shenoy, Chahak Srivastava, Chaithanya Shadakshari]

Department of Computer Science and Engineering [RV College of Engineering]

Email: [daivikshenoy.cs23@rvce.edu.in chahaks.cs23@rvce.edu.in chaithanyas.cs23@rvce.edu.in]

Abstract—Election misinformation campaigns typically do not rely on individual falsehoods but rather employ paraphrasing, coordinated amplification, and emotional manipulation to disseminate their content. The ElectionShield Live system can be used to combat such misinformation strategies through four interconnected components, including a Narrative Intelligence Engine based on RoBERTa, an emotional manipulation detection tool, an SBERT-based semantic clustering engine, and a coordination detector based on behavioral anomalies. These tools produce results that are accessible via a light-weight dashboard interface that election observers, analysts, and media organizations can use to monitor misinformation attacks. The evaluation shows classification accuracy of 91.4%, clustering F1-score of 0.887, and coordination detection accuracy of 88.2%. Moreover, the end-to-end response time is less than 2.3 seconds under realistic workloads.

Index Terms—Election Security, Coordinated Misinformation Detection, Narrative Intelligence, Semantic Clustering, Propaganda Analysis, Emotional Manipulation, Bot Detection, AI Governance, NLP, Real-Time Dashboard

I. INTRODUCTION

Political communication on digital media has moved to environments where lies may be paraphrased, amplified, and manipulated emotionally on a large scale [1], [2]. In election contexts, this results in misinformation as an election-wide problem, not just a collection of individual posts.

There are 3 major features that can be observed. Firstly, a single lie in a post can affect and change the surface level appearance in many ways [3]. Secondly, posts usually appear on the feed at the same time across various accounts and making the fake news more visible to the audience [4]. Finally, there's emotional manipulation, which takes advantage of the user's feelings like fear and anger to speeden up the process of spreading misinformation while they don't think about it as much [5].

Older methods for catching fake news do not consider these factors, and modern methods generally don't prevent or predict rather react after it has happened[6]. Thus, it takes longer to spot misinformation, a problem seen in important elections around the world, from the US and India to Europe and Brazil [7], [8]. To understand and keep up with these, election

monitors need options that understand content, emotions, and coordination all at once and also in realtime.

ElectionShield Live closes this gap with an AI system combining narrative understanding, emotional intelligence analysis, and coordination detection in one realtime system. This idea thus offers two key contributions: a contextual engine to score misinformation and propaganda within these posts, and a module to track these false and twisted narratives using semantic clustering.

- A contextual Narrative Intelligence Engine for misinformation and propaganda scoring.
- A semantic clustering module that tracks paraphrased narrative variants.
- An emotional manipulation analyzer focused on election-relevant affective signals.
- A coordination detector for synchronized and anomalous campaign behavior.
- A real-time dashboard that exposes actionable monitoring outputs.

II. LITERATURE SURVEY

A. Recent Research Papers (2023–2025)

a) *“TELLER: A Trustworthy Framework for Explainable, Generalizable and Controllable Fake News Detection” — Findings of the Association for Computational Linguistics: ACL, 2024 [25].*: This paper addresses the problem of identifying believable fake news in a domain shift setting. This task employs a two-layer approach consisting of LLM-based logic atom generation and decision-making through rules. The novelty of the work lies in its focus on interpretable and controllable reasoning, rather than pure black box classification. A weakness of the paper is that it is post-level and does not consider coordinated propagation, narrative clusters, or real-time monitoring.

b) *“HQP: A Human-Annotated Dataset for Detecting Online Propaganda” — Findings of the Association for Computational Linguistics: ACL, 2024 [24].*: The strength of HQP lies in the quality of propaganda detection due to increased

supervision. This work presents a big dataset of manually labeled propaganda texts and examines the modern language models trained on high-quality labels. What makes this work stand out is the experimental proof of improved reliability of propaganda detection due to better annotation quality. The weakness is the lack of an actual operation system for monitoring elections.

c) *“EU DisinfoTest: a Benchmark for Evaluating Language Models’ Ability to Detect Disinformation Narratives” — Findings of the Association for Computational Linguistics: EMNLP, 2024 [27].*: In contrast, this paper assesses how effectively language models recognize disinformation narratives as opposed to individual claims of misinformation. This particular benchmark has been created through human-in-the-loop curation and explicitly incorporates rhetorical components such as Logos, Pathos, and Ethos. The primary strength of this research lies in its rhetorical awareness with regard to the assessment of narrative-based misinformation recognition models. The primary weakness of this study is the fact that it provides a benchmark for models but not real-time clustering and coordination analysis.

d) *“DiNaM: Disinformation Narrative Mining with Large Language Models” — Proceedings of EMNLP, 2025 [28].*: The DiNaM approach focuses on the problem of narrative mining from disinformation analysis. First, it identifies misinformation using LLMs, and then it uses clustering to detect higher-order disinformation narratives. The primary novelty of DiNaM is its pipeline for narrative discovery along with a new narrative mining evaluation metric. The weakness of DiNaM is that it does not take into account coordinated accounts, analyst processes, or alerts. In ElectionShield Live, the most similar approach is the semantic clustering using SBERT embeddings.

e) *“Using Causality to Infer Coordinated Attacks in Social Media” — Proceedings of the International AAAI Conference on Web and Social Media, 2025 [26].*: The present paper deals with coordinated attack detection using temporal dependency, rather than content dependency alone. The technique used here is Convergent Cross Mapping together with topic modeling, in order to detect coordination at both the leader-follower level and the community level. The key strength of the paper lies in the presence of stronger behavioral indicators of manipulation in coordinated activities, such as those by the IRA. Its weakness is the lack of content explainability and a mis/disinformation-emotion stack.

Weighted relevance discussion: With an implementation-oriented weighting approach to classify trustworthy (0.25), propaganda/rhetoric awareness (0.20), semantic narration (0.20), coordination (0.20), and analyst readability (0.15), the TELLER system and DiNaM framework have the most significant architectural impacts, HQP enhances supervision the most, EU DisinfoTest improves rhetoric-awareness analysis the most, and the causality paper provides the best coordination signal. [24]–[28].

B. Research Gap

The existing frameworks, however, mostly focus on optimizing individual aspects such as fake news detection, data set optimization, benchmarking, or offline coordinated activity analysis rather than holistic real-time election intelligence analysis [24]–[28]. Few frameworks leverage semantic clustering, emotional manipulation detection, and coordinated activity detection together, and there is a lack of explainability along with real-time election-risk monitoring by analysts. ElectionShield Live tries to address these shortcomings by combining narrative intelligence through RoBERTa with SBERT semantic clustering, emotional manipulation detection, and coordinated activity detection.

III. PROBLEM STATEMENT

A. Nature of Modern Electoral Misinformation

Misinformation in elections is basically about spreading altered stories to reach political aims [23]. Fake narratives can take different forms by rewording and making it super hard to identify them with simple methods, hence the vocabulary changes easily.

B. Limitations of Existing Approaches

Current approaches are very spread out: fake news classifiers only look at posts one by one and the sentiment tools don’t check for hidden motives behind the posts, and the bot detectors just focus on single accounts rather than multiple accounts. Thus the batchwise analytics show up very late for any action to be taken at that time.

C. Research Problem Definition

For a real-time stream of election-related content, the system needs to do five main things. Firstly, it must classify each post as misinformation with understandable confidence scores. Secondly, similar narratives need to be grouped together no matter how different the words used are. Thirdly, the system should measure different levels of emotional manipulation. Fourthly, it has to spot accounts that show synchronized behavior and suggest coordinated spread. Finally, all this information needs to be shown on an interactive dashboard for anyone to access.

IV. PROPOSED SYSTEM

A. Overall Architecture

ElectionShield Live is organized into six functional layers that process incoming social media content from ingestion through visualization. Figure 1 illustrates the complete system architecture.

The pipeline ingests live posts, scores misinformation and emotion, groups semantically similar claims, checks for synchronized amplification, and publishes analyst-facing alerts.

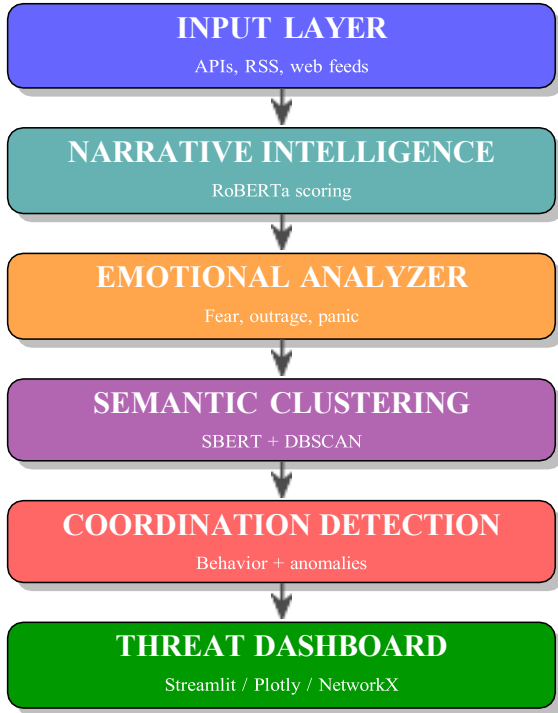


Fig. 1. ElectionShield Live System Architecture

B. Narrative Intelligence Engine

The Narrative Intelligence Engine (NIE) employs the fine-tuned **RoBERTa-base** model for per-post content analysis. RoBERTa was chosen because of its excellent context understanding regarding political content and effective identification of implicit misinformation as compared to other traditional techniques that employ keywords for identifying misinformation.

1) *Preprocessing Pipeline*: Each social media post undergoes the following preprocessing steps:

- 1) URLs and mentions are replaced with [URL] and [USER] tokens.
- 2) Compound hashtags are segmented into meaningful words.
- 3) Emojis are converted into descriptive text.
- 4) Input text is truncated to 512 tokens.

2) *Transformer-Based Narrative Detection*: The NIE uses two classification heads over a shared RoBERTa encoder:

- **Misinformation Head**: Binary classifier trained on LIAR-PLUS [12] and MultiFC [13] datasets to distinguish credible and misleading content.
- **Propaganda Head**: Multi-class classifier trained on the SemEval-2020 Propaganda Corpus [14] to identify propaganda techniques such as fear appeal, loaded language, and false dilemma.

RoBERTa differs from keyword-based filters in that it understands the context and semantics of election-themed disinformation such as “the election was fixed” or “the voting machines have been hacked.”

Every message has associated scores $c \in [0, 1]$. When the score for misinformation goes above 0.65, the message is flagged as **HIGH_RISK** and passed on to the downstream verification modules.

3) *Explainability*: In order to increase transparency, the NIE selects the top five tokens that contribute to attention generation from the transformer model and outputs them along with the classification outputs.

Algorithm 1 summarizes the NIE workflow.

Algorithm 1 Narrative Intelligence Engine Processing

Require: Post p , RoBERTa model M , threshold $\tau = 0.65$

```

1:  $p' \leftarrow \text{Preprocess}(p)$ 
2:  $h \leftarrow M_{\text{encoder}}(p')$ 
3:  $C_{\text{mis}} \leftarrow \text{Softmax}(W_1 h_{[CLS]})$ 
4:  $C_{\text{prop}} \leftarrow \text{Softmax}(W_2 h_{[CLS]})$ 
5: tokens  $\leftarrow \text{TopK}(\text{AttentionWeights}, 5)$ 
6: if  $C_{\text{mis}} > \tau$  then
7:   flag  $\leftarrow \text{HIGH\_RISK}$ 
8: else
9:   flag  $\leftarrow \text{NORMAL}$ 
10: end if
11: return  $\{C_{\text{mis}}, C_{\text{prop}}, \text{tokens}, \text{flag}\}$ 
  
```

C. Emotional Manipulation Analyzer

The Emotional Manipulation Analyzer (EMA) aims at detecting propaganda that is emotion-oriented. Emotionally charged arguments, particularly those that instill feelings of fear, outrage, and/or panic, have been widely employed in the process of influencing opinion and hindering rational political discourse. [10].

1) *Emotional Manipulation Detection*: The EMA relies on a well-trained **RoBERTa-large** architecture based on the GoEmotions corpus [15] in addition to a new labeled data set of political social media messages. This classification system is based on six emotions, including *Fear*, *Outrage*, *Panic*, *Disgust*, *Hope*, and *Neutral*. Unlike common sentiment analysis approaches, the EMA focuses on detecting emotions associated with manipulation,

2) *Manipulation Index Computation*: For each post p , the EMA generates an emotion probability vector:

$$\mathbf{e} = [e_{\text{fear}}, e_{\text{outrage}}, e_{\text{panic}}, e_{\text{disgust}}, e_{\text{hope}}, e_{\text{neutral}}] \quad (1)$$

The *Emotional Manipulation Index* (EMI) is computed as:

$$\text{EMI}(p) = \alpha e_{\text{fear}} + \beta e_{\text{outrage}} + \gamma e_{\text{panic}} + \delta e_{\text{disgust}} \quad (2)$$

where $\alpha = 0.35$, $\beta = 0.30$, $\gamma = 0.25$, and $\delta = 0.10$ are empirically selected weights based on political communication studies [5]. Fear and outrage carry more weight because they strongly link to election distrust and coordinated propaganda campaigns. Posts with an EMI above 0.70 get identified as High Manipulation Risk and appeared in the system dashboard. When groups of posts keep having high EMI values, it is a sign of possible organized propaganda. This helps differentiate planned manipulation from regular political disagreements.

3) *Narrative Toxicity Analysis*: The Emotional Manipulation Analysis (EMA) goes beyond by figuring out cluster-based emotional profiles rather than just looking at post scores. After detecting a semantic cluster, the EMA calculates the average EMI value and the main emotion category of all accounts within that cluster. High EMI values across clusters of emotionally similar posts can show organized attempts to influence public opinion. Instead of focusing only on individual posts, the system examines broader trends in online discussions. Not only does this help detect fear in a single message, but it also shows if many users are spreading versions of a political story based on fear.

D. Semantic Narrative Clustering Engine

SCE, is a core component of ElectionShield Live, rather than relying only on exact word matches, it can recognize when different posts convey the same underlying message, even if they are written in different ways.

1) *Embedding Generation*: We use the SBERT with the all-mpnet-base-v2 model [12] to create 768-dimensional semantic embeddings. SBERT understands the whole meaning of a sentence, rather than just individual words. This makes it different from token-based embeddings and making it great for spotting rewritten narratives. Every post gets turned into a 768-dimensional vector. To find out how similar two posts are, we use cosine similarity:

$$\text{sim}(p_i, p_j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \|\mathbf{v}_j\|} \quad (3)$$

2) *Semantic Narrative Intelligence*: The SCE doesn't just focus on exact text matches, it looks for shared narrative intent instead. For example, discussing election fraud or mistrust in institutions may use different wordings, yet the system can spot the same claim being spread. This allows the SCE to group related misinformation into clusters based on their meaning, not just their words.

3) *Clustering Algorithm*: The SCE uses DBSCAN clustering with cosine distance to cluster posts based on their semantic similarity. They choose DBSCAN for a few reasons: it works well with the steady and continuous flow of social media data. It doesn't need a predetermined number of clusters, it can naturally separate unrelated or noisy posts from meaningful discussions. Plus, the params = 0.25 and minPts = 3 were chosen after trial and error to grab narrative variations without grouping everything together in huge clusters.

4) *Narrative Mutation Tracking*: Each cluster gets a special narrative ID. As new posts come in, the system checks whether they fit within an existing cluster or form a new narrative. If a cluster's centroid moves a lot between time windows, the system can detect narrative mutation events, which signal shifts in the message being spread. This helps ElectionShield Live follow changes in messaging – think moving from general election skepticism about elections to specific claims about vote machines or biases.

For a peek at how it works, check Table I which shows different posts being grouped into one narrative cluster based on their deeper meaning.

TABLE I
EXAMPLE: SEMANTIC CLUSTERING OF ELECTORAL MISINFORMATION VARIANTS

Post Text	Cluster ID
"The election results are totally fake"	Cluster-07
"votes were changed by people behind the scenes"	Cluster-07
"The electronic counting machines got hacked"	Cluster-07
"no one trusts the official vote count"	Cluster-07

5) *Graph Visualization Layer*: The SCE uses NetworkX to create a real-time graph on their dashboard. It shows how misinformation spreads. In this graph, each post is a node, and the edges show how similar they are. Also, the color of each node tells you which group the post is part of, and its size reflects a high EMI score. This helps analysts spot misinformation quickly.

E. Coordination Detection Engine

The Coordination Detection Engine looks at how posts are shared across multiple accounts and detects coordinated amplification activity. Here's how it works: It checks each account in a set time frame, usually 30 minutes. It looks at posting frequency and patterns, the average gap between post times, how related the content is to risky topics, how similar hashtags are used, and the rate of duplicated content based on the cosine similarity > 0.85.

1) *Anomaly Detection with Isolation Forest*: Anomaly detection uses an Isolation Forest model for detecting abnormal behavior in the collected data. This model is perfect for big data flow and it finds irregularities all by itself—no labeled examples needed. Accounts scoring below -0.5 are marked as Coordination Suspicious Actors (CSA). When there are more than three CSAs expanding the same content in within 30 minutes, it triggers a Coordinated Campaign Alert. By checking both strange behaviors and what users are talking about, along with timing, they reduce the issue of wrongly identifying actual legitimate high-activity accounts.

2) *Coordinated Influence Detection*: Next is the coordinated influence detection. For this, they use a Random Forest classifier which has got a 100 estimators. It does a secondary validation check using well known labeled data sets like the Internet Research Agency and COVID-19 Twitter Coordination Data set. This helps assign coordination probability scores and types, like State Actor, Political Group, Regular User Boost, or Bot networks. The two-stage design helps make the detection result more accurate and reliable. Isolation Forest detects odd behaviors, and Random Forest figures out the probable coordination behind them. It's really helpful during elections because synced postings, repeated hashtags, and semantic duplication usually show up before big campaigns. Algorithm 2 sums up the CDE workflow.

Algorithm 2 summarizes the CDE workflow.

Algorithm 2 Coordination Detection Engine

Require: Account posts P , time window Δt , SCE clusters C

```

1: for each account  $a$  in  $P$  do
2:    $f_a \leftarrow \text{ExtractFeatures}(a, \Delta t, C)$ 
3: end for
4:  $s \leftarrow \text{IsolationForest.score}(\{f_a\})$ 
5:  $CSA \leftarrow \{a : s_a < -0.5\}$ 
6: for each cluster  $C$  in  $C$  do
7:    $CSA_c \leftarrow \{a \in CSA : \text{posts in } C\}$ 
8:   if  $|CSA_c| \geq 3$  then
9:     label  $\leftarrow \text{RandomForest.predict}(\{f_a : a \in CSA_c\})$ 
10:    A.append(Alert( $C, CSA_c, \text{label}$ ))
11:   end if
12: end for
13: return A

```

F. Interactive Threat Intelligence Dashboard

The visualization tools include Streamlit, Plotly, FastAPI, and NetworkX/PyVis. Instead of a standard analytics dashboard, we build our dashboard to act as a real-time election intelligence surface. The core reason behind this design is latency – once a negative story has been manually identified in a batch process, there might be no way to intervene because of how fast the coordinated amplification efforts normalized the harmful narrative. Hence, the dashboard displays live data signals for individual posts, clusters, and campaigns.

Three primary views of an analyst's workspace include:

- Live narrative graph showing active clusters and mutations,
- Cluster and emotion inspection view, and
- Coordination timeline, including JSON and PDF alert notifications.

All three views combine into an effective toolset for election intelligence operations. An analyst may verify whether the story is spreading semantically, whether emotional engineering was involved in spreading it, and whether accounts showed synchronizing behavior. In this context, our solution addresses the weaknesses inherent in moderated systems.

V. IMPLEMENTATION METHODOLOGY

A. System Architecture and Technology Stack

The ElectionShield Live System utilizes a modular approach where each stage of analysis is individually scalable. Instead of an offline processing model, the design objective focuses on minimizing latency for campaign inference. This is shown in Figure 2.

B. End-to-End Dataflow

The processing pipeline is organized as the following service chain:

- 1) **API ingestion:** election-related posts are collected from approved APIs or feed connectors and normalized into a common JSON event schema.
- 2) **Preprocessing:** URLs, mentions, hashtags, timestamps, and account metadata are cleaned and standardized before inference.

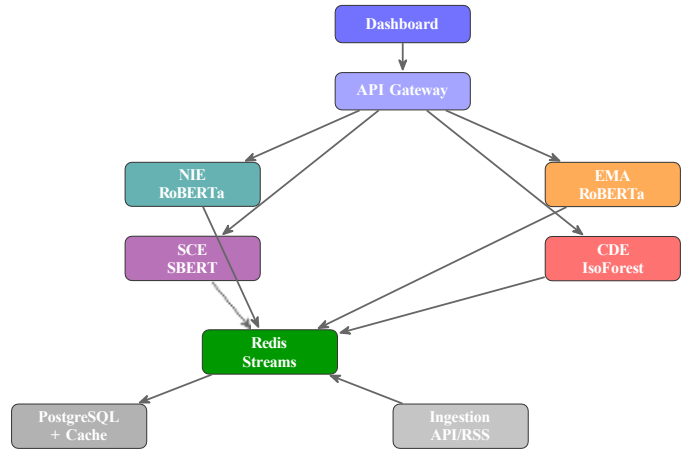


Fig. 2. ElectionShield Live Deployment Architecture

- 3) **Narrative inference:** the NIE assigns misinformation and propaganda scores and returns token-level explanations.
- 4) **Emotion inference:** the EMA generates emotion probabilities and an Emotional Manipulation Index for the same post batch.
- 5) **Vector encoding and clustering:** the SCE embeds incoming posts, retrieves candidate neighbors through FAISS, and updates DBSCAN narrative clusters.
- 6) **Behavioral coordination scoring:** the CDE aggregates account behavior over time windows and emits anomaly-based campaign alerts.
- 7) **Persistence and caching:** enriched records, cluster metadata, and alert histories are written to PostgreSQL and low-latency cache layers.
- 8) **Dashboard serving:** Streamlit requests aggregated views from the FastAPI backend for analyst triage, export, and audit.

C. Backend Implementation and API Workflow

ElectionShield Live uses Python 3.11 for its backend in a microservice setup along with a FastAPI control layer. FastAPI handles receiving data, handling user requests, and serving analyst endpoints, but it sends heavy computing tasks to separate worker services.

A typical processing cycle begins when election-related posts are received through endpoints such as `POST /ingest/posts`. For asynchronous processing, the received data is converted into a common JSON format and pushed into Redis Streams. Worker services consume the stream, execute the NIE and EMA inference pipelines, and generate enriched events that are forwarded to the clustering and coordination analysis modules.

This asynchronous design makes the system much more responsive during peak activity times. FastAPI acts as a central controller, handling user requests, authentication, and transforming API responses, while Redis Streams offers buffering, durability, and control over back-pressure between different stages. Rather than tightly coupled internal calls, the services

TABLE II
MODULE-LEVEL METHODOLOGY AND SERVICE INTERFACES

Module	Input	Processing	Output	Storage	Service Interface
Ingestion Layer	API/feed timestamps, metadata	payloads, account normalization, rate-limit aware batching, source tagging	normalized records	transient queues, raw audit log	FastAPI ingest routes, internal workers
NIE	normalized text batches	tokenization, RoBERTa inference, propaganda head scoring	misinformation score, propaganda label, explanation tokens	PostgreSQL inference table	internal REST / async inference workers
EMA	normalized text batches	emotion inference, EMI computation, cluster-level aggregation hooks	emotion vector, EMI, dominant emotion	PostgreSQL emotion table	internal REST / async inference workers
SCE	high-risk posts, embedding requests	SBERT encoding, FAISS neighbor retrieval, DBSCAN update	cluster ID, centroid, mutation flag	vector cache, cluster tables	internal RPC/REST plus Redis event stream
CDE	account-window events, cluster assignments	feature extraction, Isolation Forest scoring, RF campaign typing	CSA set, alert probability, campaign label	alert history tables, cache	scheduled workers plus alert API
Dashboard	aggregated cluster, alert, and metric views	filtering, visualization, export formatting	live graph, heatmap, timeline, reports	cached query layer	Streamlit frontend over FastAPI

use structured message schemas, hence the modules can be independently updated, scaled, or redeployed without disrupting the whole system.

The backend is organized into these service components: - The Ingestion Layer gathers election-related data from APIs and RSS feeds and stores it in Redis Streams. - The NIE and EMA Services do batched transformer-based narrative and emotional analysis using FastAPI worker endpoints. - The SCE Service creates SBERT embeddings, does similarity searches with FAISS, and performs sliding-window DBSCAN clustering. - Lastly, the CDE Service aggregates behavioral features, detects anomalies, and manages coordination classification.

D. Storage, Retrieval, and Service Communication

Redis Streams serves as the short-horizon transport layer because it can handle append-only event processing, consumer groups, and recoverable offsets during bursty election traffic. PostgreSQL then stores enriched post records, cluster assignments, campaign alerts, audit metadata, and analyst actions. FAISS acts as the semantic retrieval layer to let newly arriving posts get compared against recent embedding neighborhoods before DBSCAN updates.

This setup clearly separates concerns. Redis deals with moving events in real-time, PostgreSQL offers reliable history for reporting and audits, and FAISS allows quick nearest-neighbor lookups for semantic similarity tasks. In the end, content, behavior, and historical alert states stay easily queryable without bogging down the inference workers.

E. Frontend and Analyst Workflow

Streamlit's frontend relies on shared session state, FastAPI polling, and responsive Plotly views for both analyst workstations and wall-display monitoring. Typically, an analyst's session starts with a cluster summary query, then looks at token-level and emotional explanations, and finally ends with either a report being exported or an alert for a coordinated

campaign being escalated. So, the UI operates more as an operational triage surface than just a static reporting page.

F. Model Fine-Tuning Details

For the NIE, it's fine-tuned from roberta-base using LIARPLUS, MultiFC, and SemEval-2020 Propaganda datasets with AdamW at 2×10^5 and a batch size of 32. The EMA, on the other hand, is fine-tuned from roberta-large on GoEmotions and a custom political corpus, again with AdamW, but at 1×10^5 . Doing it this way keeps the tasks specific while keeping things fast enough for actual use. During deployment, tokenization and input assembly run on CPU workers; batches get sent to the GPU memory, and inference outputs come back as compact JSON payloads, helping to cut down on the time needed for serialization.

G. Deployment Logic and Scalability

The system runs in Docker Compose and can handle around 800 posts p/m on a single server with 32GB RAM, 8 cores and an NVIDIA T4 GPU, while keeping end to end latency under 3 seconds. Since ingestion workers, inference services, and dashboard processes are separate components, it scales horizontally easily. During high load; FastAPI replicas behind a reverse proxy, expand Redis consumer groups, and replicate GPU-backed inference containers independently from the dashboard and storage services can be added. This setup makes the process much easier to deploy. With election related traffic spiking during debates, candidate incidents, polling day, and counting periods, having a modular pipeline that uses asynchronous messaging, vector retrieval, GPU batching, and cached analyst views is way more practical than trying to run a single synchronous app for everything.

VI. INTELLECTUAL PROPERTY RIGHTS

A. Secure API-Based Data Acquisition

ElectionShield Live gathers election related content exclusively from authorized sources like public APIs or approved

feed integrations. This method keeps things legal and compatible with terms of service, API policies, and rate limits [35], [36]. They use authenticated API access, scoped tokens, and controlled request throttling to make sure data collection is follows all the rules and thus becomes secure.

Secure APIs also shields their process from issues related to intellectual property protection. Publicly visible posts can still have copyright rules with redistribution restrictions. By collecting information only from official APIs, ElectionShield stays clear of violating third party rights and makes it easier to track content origins. Also, the encrypted communications, safe storage for credentials and rotating tokens boost the security globally.

B. Privacy Preservation and Data Protection

ElectionShield Live is meant for monitoring to preserve election integrity, not for censorship or political targeting. It protects privacy and the framework keeps personally identifiable information minimum throughout its process. Things like usernames, account IDs, and sensitive metadata are either hashed, masked, or replaced with pseudo IDs before being stored. So instead of showing individual users, the dashboard highlights narrative trends, emotional manipulation, and coordination patterns. The system includes encrypted storage and role-based access control to boost data security [33]. While raw posts might be needed temporarily during analysis, there's no permanent leakage of sensitive information once analytical features are made.

C. Intellectual Property Protection of the Framework

ElectionShield Live has plenty of protectable features that are beyond the research paper. This can include models like their trained RoBERTa and SBERT models, ways to cluster information syntactically, spot coordination, score emotional manipulation, set up dashboards and gather data tailored for elections.

They could opt to protect their source code, design interfaces build by them, and document their technology using software copyrights. Branding and product names usually get covered by trademarks too.

Some parts which are like model weights, decision boundaries, clustering rules, and setup settings, work best under these confidentiality as trade secrets [32].

While public social media content isn't private always and things like labeled data, embeddings, notes and smart reports can still get protection under it.

D. Disclosure, Licensing, and Institutional Deployment Safety

The security of ElectionShield Live depends on more than just intellectual property ownership. It relies also on careful handling of information. While patents protect unique methods, they demand a public disclosure that could harm the confidentiality associated with it. But using Software as a Service (SaaS) and licensed API access allows the team to share the final results and not the crux of how those analysis happen. To attain equilibrium, a mixed approach is

the best. Academic honesty needs detailed methodologies to be present for scrutiny. Yet, things like trained models, security controls, and specific setups tailored for an institution must stay under wraps. Considering the safest path it's all about managing who gets access through licensed applications. Not everything needs to be a wide open deal, or either it should be. Concrete licensing agreements covering terms like; ownership, use, audit jabs, data limits, and re-sharing rights will protect operations while keeping academics satisfied with what they need.

VII. EXPERIMENTAL RESULTS

A. Experimental Setup

We use three different datasets for our experiments; 1) LIAR-PLUS [25] (looks at verified political statements); 2) CoAID [34], (for out-of-domain misinformation samples to check robustness); 3) ElectionShield-Eval, (set of 5,000 election-related posts checked for misinformation, emotion and coordination cues).

B. Misinformation Classification Performance

Table III reports classification performance of the NIE across datasets.

TABLE III
NIE MISINFORMATION CLASSIFICATION PERFORMANCE

Dataset	Accuracy	Precision	Recall	F1
LIAR-PLUS	91.4%	0.912	0.916	0.914
CoAID	86.3%	0.857	0.869	0.863
ElectionShield-Eval	89.2%	0.888	0.897	0.892
Average	89.0%	0.886	0.894	0.890

C. Semantic Clustering Evaluation

They evaluated the SCE using homogeneity, completeness. The V-measure against narrative groups labelled manually in ElectionShield Eval

TABLE IV
SCE SEMANTIC CLUSTERING PERFORMANCE

Method	Hom.	Comp.	V-Meas.	Sil.
TF-IDF + K-Means	0.721	0.698	0.709	0.541
BERT + K-Means	0.804	0.791	0.797	0.612
SBERT + DBSCAN	0.891	0.883	0.887	0.731

The SBERT+DBSCAN setup beat both baselines by a great margin, meaning that semantic embedding works well for detecting narrative mutations.

D. Emotional Manipulation Analysis

Table V reports selected election-relevant emotion classes.

E. Coordination Detection Performance

Table VI reports CDE performance on 250 manually labeled coordination scenarios from the Internet Research Agency dataset [17].

TABLE V
EMA EMOTION CLASSIFICATION PERFORMANCE PER CLASS

Emotion	Precision	Recall	F1	Support
Fear	0.883	0.871	0.877	412
Outrage	0.896	0.904	0.900	538
Panic	0.851	0.843	0.847	287
Neutral	0.943	0.951	0.947	953
Weighted Avg	0.899	0.904	0.901	3000

TABLE VI
CDE COORDINATION DETECTION PERFORMANCE

Method	Accuracy	Precision	Recall	F1
Rule-based	71.2%	0.684	0.741	0.711
RF only	84.7%	0.839	0.855	0.847
IF only	81.3%	0.801	0.826	0.813
IF + RF (proposed)	88.2%	0.874	0.891	0.882

F. System Performance and Latency

Table VII reports representative end-to-end latency under increasing load.

TABLE VII
SYSTEM PERFORMANCE UNDER SIMULATED LOAD

Load (posts/min)	NIE (ms)	SCE (ms)	CDE (ms)	E2E (s)
100	82	47	31	0.91
800	411	219	148	2.31
1200	743	388	261	4.18

The system remains below 3 seconds at 800 posts/minute, which is sufficient for a single-platform election-monitoring stream. Latency rises under heavier burst load but remains operational.

VIII. ETHICAL AND SOCIAL IMPLICATIONS

A. Human-in-the-Loop Review and Explainability

ElectionShield Live is an analysis aid system as opposed to an automation system for moderation. The confidence score, explanation per token, cluster origin, and history of alerts were created for human review and not to replace human review altogether. This is especially critical during elections when false positives might hamper political engagement and even true alerts could be interpreted through context.

B. Bias Mitigation and Multilingual Fairness

A bias threat is one that persists irrespective of the language, geographical locations, and political environment. A model trained mostly on English political datasets will have problems when handling other languages, regionalisms, slang, and sarcasm. Some ways through which such a problem can be addressed include using multilingual evaluation datasets, conducting audits of political biases, calibrating thresholds depending on the domain, and providing error rates based on language, topics, and political environment. It is only through this approach that we will enhance elections' credibility.

C. Misuse Prevention and Democratic Safeguards

This infrastructure, which is able to detect manipulation that is coordinated, can be used for surveillance or for enforcement, in an undemocratic manner. This is why the governance model of the project must include clear usage guidelines: while it would be used to monitor and fight misinformation in elections and in other matters of public interest, it shouldn't be used for censorship, suppression of the vote, or any other purpose that could interfere in any way with democracy.

D. Transparency, Accountability, and Contestability

The other way of ensuring accountable AI governance is to ensure that the affected organizations will be able to determine the reasoning behind the triggering of an alarm. ElectionShield Live will gain from the use of model cards, recorded version numbers of parameters, auditable trail of alerts, and storage of the evidence considered while making every coordination action. Where possible, the contested outputs need to be subject to revision. This governance approach ensures that the system gains credibility in a high-stakes environment such as elections.

IX. FUTURE SCOPE

Near-term extensions include multilingual modeling, cross-platform narrative correlation, predictive alerting, and tighter integration with election-monitoring workflows. These directions preserve the current architecture while expanding geographic coverage and operational value.

X. CONCLUSION

This paper presents **ElectionShield Live**, a real-time framework for identifying semantically adaptive, emotionally manipulative, and behaviorally coordinated election narratives. By combining narrative classification, emotion scoring, semantic clustering, coordination detection, and dashboard visualization, the system moves beyond isolated fake-news prediction toward campaign-level monitoring.

Results indicate strong performance across the main tasks, including 91.4% misinformation classification accuracy, 0.887 clustering F1, 0.901 weighted emotion F1, and 88.2% coordination detection accuracy, while maintaining near-real-time response. So, the framework keeps the paper's main technology more usable form.

REFERENCES

- [1] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu, "Combating Fake News: A Survey on Identification and Mitigation Techniques," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 3, pp. 1–42, 2019.
- [2] R. DiResta, K. Shaffer, B. Ruppel, D. Sullivan, R. Matney, R. Fox, J. Albright, and B. Johnson, "The Tactics and Tropes of the Internet Research Agency," New Knowledge, Tech. Rep., 2019.
- [3] V. Pe'rez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic Detection of Fake News," in *Proc. 27th Int. Conf. Comput. Linguistics (COLING)*, 2018, pp. 3391–3401.
- [4] O. Varol, E. Ferrara, C. A. Davis, F. Menczer, and A. Flammini, "Online Human-Bot Interactions: Detection, Estimation, and Characterization," in *Proc. 11th Int. AAAI Conf. Web Social Media (ICWSM)*, 2017, pp. 280–289.

- [5] W. L. Bennett and S. Livingston, "The Disinformation Order: Disruptive Communication and the Decline of Democratic Institutions," *Eur. J. Commun.*, vol. 33, no. 2, pp. 122–139, 2018.
- [6] C. Wardle and H. Derakhshan, "Information Disorder: Toward an Interdisciplinary Framework for Research and Policymaking," Council of Europe, Tech. Rep. DGI(2017)09, 2017.
- [7] P. N. Howard, S. Bradshaw, B. Kollanyi, and L.-M. Neudert, "Sourcing and Automation of Political News and Information During Three Simultaneous Elections," *Political Commun.*, vol. 35, no. 2, pp. 331–344, 2018.
- [8] R. Evangelista and F. Bruno, "WhatsApp and Political Instability in Brazil: Targeted Messages and Political Radicalisation," *Internet Policy Rev.*, vol. 8, no. 4, 2019.
- [9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," in *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2019, pp. 3982–3992.
- [10] W. J. Brady, J. A. Wills, J. T. Jost, J. A. Tucker, and J. J. Van Bavel, "Emotion Shapes the Diffusion of Moralized Content in Social Networks," *Proc. Natl. Acad. Sci.*, vol. 114, no. 28, pp. 7313–7318, 2017.
- [11] H. Allcott and M. Gentzkow, "Social Media and Fake News in the 2016 Election," *J. Econ. Perspect.*, vol. 31, no. 2, pp. 211–236, 2017.
- [12] W. Y. Wang, "'Liar, Liar Pants on Fire': A New Benchmark Dataset for Fake News Detection," in *Proc. 55th Annu. Meet. Assoc. Comput. Linguistics (ACL)*, 2017, pp. 422–426.
- [13] I. Augenstein, C. Lioma, D. Wang, L. C. Lima, C. Hansen, J. Hansen, and J. Simonsen, "MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking," in *Proc. EMNLP-IJCNLP*, 2019, pp. 4685–4697.
- [14] G. Da San Martino, A. Barro'n-Ceden'o, H. Wachsmuth, R. Petrov, and P. Nakov, "SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles," in *Proc. 14th Workshop Semantic Eval. (SemEval-2020)*, 2020, pp. 1377–1414.
- [15] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "GoEmotions: A Dataset of Fine-Grained Emotions," in *Proc. 58th Annu. Meet. Assoc. Comput. Linguistics (ACL)*, 2020, pp. 4040–4054.
- [16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.
- [17] D. Lukito, J. Storygard, F. Yang, J. Yoder, C. Wells, and J. C. M. Peng, "The Wolves in Sheep's Clothing: How Internet Research Agency Tweets Appeared in U.S. News as Organic Social Media," *Polit. Commun.*, vol. 37, no. 2, pp. 196–216, 2020.
- [18] T. Sharma, M. Alam, F. Alam, and M. Imran, "Identification of Coordinated Inauthentic Behavior in COVID-19 Twitter Discourse," in *Proc. Int. AAAI Conf. Web Social Media (ICWSM)*, 2022, pp. 959–970.
- [19] World Intellectual Property Organization (WIPO), "Trade Secrets," WIPO Intellectual Property Portal.
- [20] Government of India, "The Digital Personal Data Protection Act, 2023," Ministry of Law and Justice, Gazette of India, 2023.
- [21] D. Hu, C. Du, H. Li, Y. Liao, A. S. M. Mosa, and I. Atkinson, "CoAID: COVID-19 Healthcare Misinformation Dataset," *arXiv preprint arXiv:2006.00885*, 2020.
- [22] D. Hardt, "The OAuth 2.0 Authorization Framework," IETF RFC 6749, Oct. 2012.
- [23] M. Jones and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage," IETF RFC 6750, Oct. 2012.
- [24] A. Maarouf, D. Ba'r, D. Geissler, and S. Feuerriegel, "HQP: A Human-Annotated Dataset for Detecting Online Propaganda," in *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand, 2024, pp. 6064–6089.
- [25] H. Liu, W. Wang, H. Li, and H. Li, "TELLER: A Trustworthy Framework for Explainable, Generalizable and Controllable Fake News Detection," in *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand, 2024, pp. 15556–15583.
- [26] I. Manchanayaka, Z. R. Zaidi, S. Karunasekera, and C. Leckie, "Using Causality to Infer Coordinated Attacks in Social Media," in *Proc. Int. AAAI Conf. Web Social Media (ICWSM)*, vol. 19, no. 1, 2025, pp. 1176–1189.
- [27] W. Sosnowski, A. Modzelewski, K. Skorupska, J. Otterbacher, and A. Wierzbicki, "EU DisinfoTest: a Benchmark for Evaluating Language Models' Ability to Detect Disinformation Narratives," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, Florida, USA, 2024, pp. 14702–14723.
- [28] W. Sosnowski, A. Modzelewski, K. Skorupska, and A. Wierzbicki, "Di-NaM: Disinformation Narrative Mining with Large Language Models," in *Proc. 2025 Conf. Empirical Methods Natural Language Processing (EMNLP)*, Suzhou, China, 2025, pp. 30224–30251.