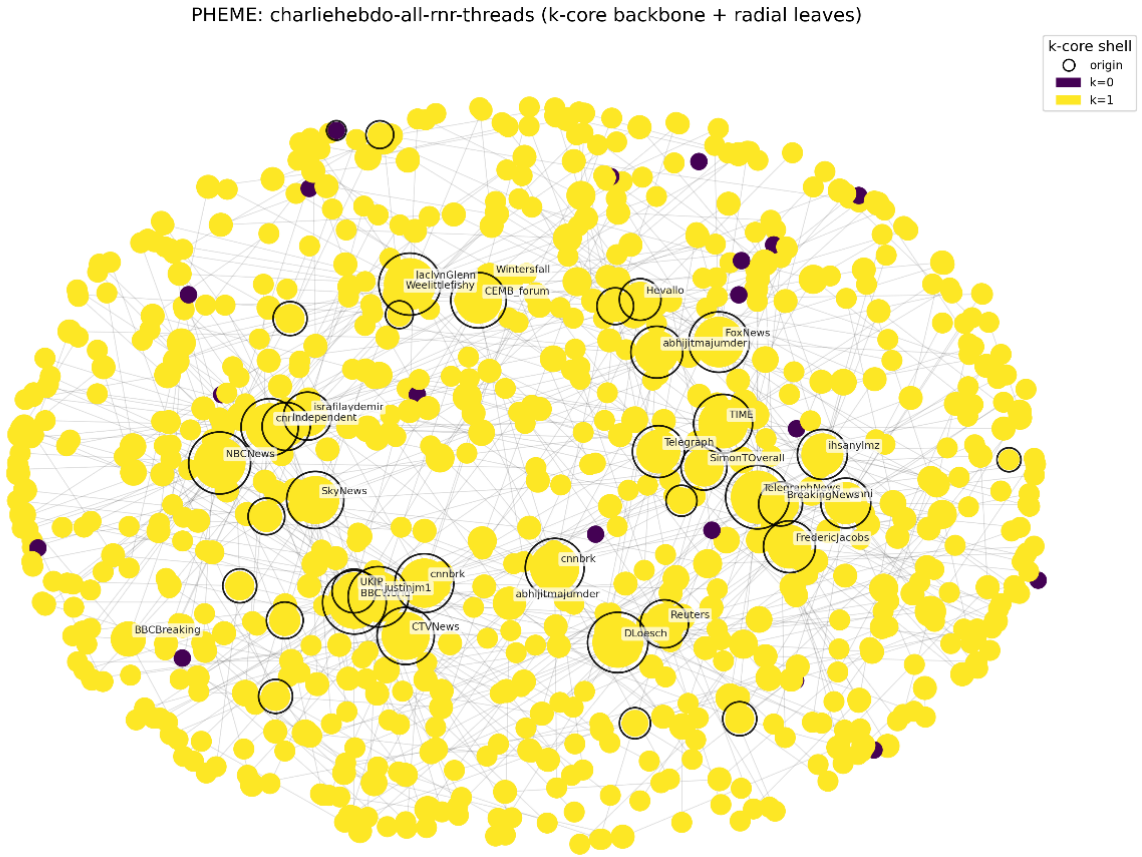
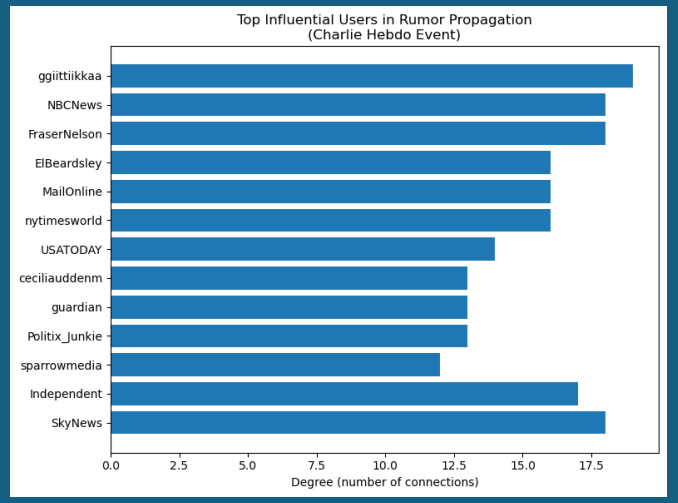
Rumor Propagation in the PHEME Dataset: Insights from the Charlie Hebdo Event

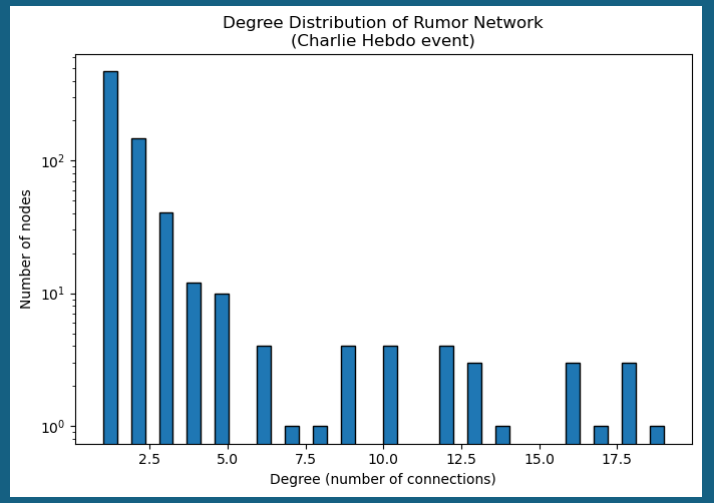


- **Nodes = Twitter users** participating in rumor or non-rumor discussions
- **Edges = Interactions** (reply / retweet) between users
- **Node size = Influence level** (degree; larger = more connected)
- **Node color = k-core shell**
- Yellow → highly connected core
- Purple → peripheral participants
- **Black outline = origin tweets** (starting points of rumor threads)

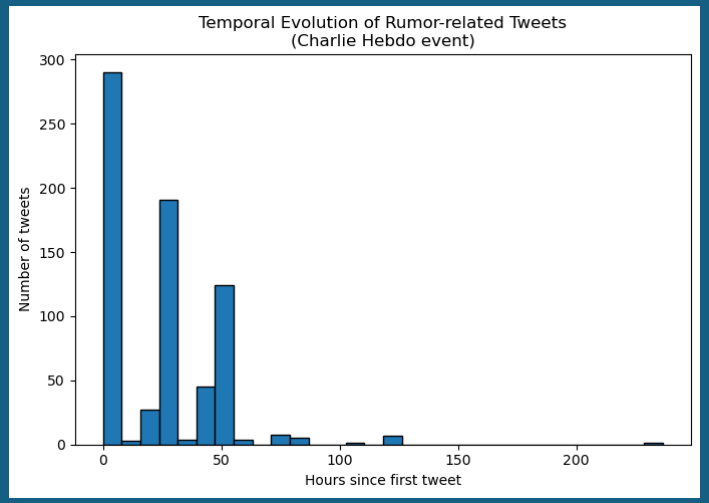
The visualization reveals a clear core–periphery structure: major media outlets such as BBCBreaking, CNN, Reuters, and SkyNews dominate the central region and serve as primary information hubs. Several independent origin tweets indicate that multiple parallel rumor conversations emerged simultaneously rather than being driven by a single viral post. Peripheral users form a large reactive audience that mainly amplifies content produced by highly connected core accounts, demonstrating asymmetric influence within crisis-related discussions.



Key media and journalist accounts occupy the highest degrees, indicating that institutional actors dominate rumor visibility and shape the flow of information. These influential nodes act as broadcast hubs that substantially amplify message reach.



The network exhibits a heavy-tailed degree distribution, meaning most users have very few interactions, while a small number of hubs maintain extremely high connectivity. This structural inequality explains why rumors can spread widely with very limited user participation.



Tweet activity spikes sharply immediately after the event, followed by rapid decline and intermittent secondary peaks. This burst-and-decay pattern reflects the typical public reaction cycle in crisis events, where attention is intensely concentrated but short-lived.

This study uses the PHEME dataset, focusing on the Charlie Hebdo event. All annotated rumor and non-rumor threads were parsed from tweet JSON files. User interactions were merged into a single graph, and k-core decomposition was applied to reveal structural cohesion. A spring layout positioned core nodes, with radial placement for peripheral nodes. Additional statistical visualizations were generated using NetworkX and Matplotlib in Python.