Web and Cloud Based Security Mid-Term Report

Rushabh Dharamshi
rd5g23@soton.ac.uk
University of Southampton
United Kingdom

Bharatraj KR
bkr1n23@soton.ac.uk
University of Southampton
United Kingdom

Osama Heweitat
oh5g23@soton.ac.uk
University of Southampton
United Kingdom

Radoslaw Kucisz
rk6g23@soton.ac.uk
University of Southampton
United Kingdom

1 Introduction

1.1 Background & Context

YouTube is one of the largest platforms for sharing content and social interaction around the world, hosting billions of daily engagements across videos, channels, and communities. Its open ecosystem empowers creators and facilitates discussion, but is increasingly exploited by automated accounts, or "bots," which distort engagement metrics through fake comments, inflated likes, and coordinated posting [1,3,4,15,16]. Such activity threatens the integrity of user interactions, promotes misinformation, and undermines trust in platform metrics, making the reliable detection of automated behaviour a critical challenge for both researchers and platform administrators [5,6].

1.2 Specific Problems

Detecting anomalous commenting behaviour on YouTube is particularly difficult because modern bots have evolved to mimic human behaviour, operate in coordinated "commenter mobs," and exploit network structures to evade content-based filters [15,16]. Conventional detection methods, including platform-level anti-spam mechanisms and supervised machine learning approaches, often fail against these sophisticated strategies and zero-day attacks where new unseen forms of bot traffic are not recognised by the supervised method [1,3,4,7]. Manually labelling bot activity is infeasible at scale [3,6,12]. Therefore, there is a pressing need for scalable, interpretable, and unsupervised techniques that can identify abnormal patterns of engagement without relying on extensive labelled data [11–14].

1.3 Brief Overview of Past Research

Previous studies have applied supervised machine learning and deep learning models, such as SVM, Random Forest, LSTM, and GRU, to detect spam and bot activity with notable success [1,3,4,7,8]. Although effective, these methods are limited by their reliance on labelled datasets, heavy preprocessing, and difficulty adapting to emerging bot behaviours [3,4,7]. Recent research has explored unsupervised and structure-aware approaches, leveraging graph embeddings, multivariate time series, and anomaly thresholding to uncover coordinated or subtle bot activity [11–16]. Despite these advances, a gap remains in providing a robust, interpretable, and scalable framework specifically tailored for detecting anomalous commenting behaviour on YouTube [14–16].

1.4 Research Questions

This project investigates the following research questions:

RQ1: How can unsupervised anomaly detection identify automated commenting behaviour on YouTube without relying on labelled data?

RQ2: Which structural and behavioural features such as posting frequency, account age, and co-commenter patterns most effectively differentiate bots from human users?

RQ3: How can the PROS technique (Pivot and Seek Rank-One Submatrix) be applied to model normal traffic and detect coordinated or anomalous patterns while remaining interpretable to human analysts?

RQ4: What YouTube channel categories and genres have the most bot traffic?

2 Related Work

2.1 Background and Scope

This review synthesizes prior work on content-based, hybrid, and unsupervised approaches, highlighting their methodologies, strengths, limitations, and the progression toward structure-aware detection.

2.2 Supervised Content-Based Approaches

Early work relied heavily on supervised machine learning using textual features extracted from YouTube comments. For instance, a study comparing classical ML methods (Logistic Regression, Random Forest, SVM, FastText) with deep learning models (LSTM, GRU) on 2,519 preprocessed comments showed that FastText achieved the highest accuracy at 96%, outperforming GRU (91.6%) [1]. Text preprocessing included tokenization, lemmatization, stopword removal, and embedding via TF-IDF and Word2Vec. Evaluation metrics included Accuracy, Precision, Recall, F1-score, and ROC AUC. These results demonstrate that traditional ML can outperform deep learning on small to medium datasets while remaining computationally efficient [1].

Similarly, Aiyar and Shetty [3] collected 13,000 comments labelled as spam or ham to train classifiers using Bag-of-Words and character-level N-gram features. Character-level N-grams with SVM achieved an F1 score of 0.984, effectively capturing obfuscated spam messages. While these supervised methods perform well, they require manually labelled data, are language-specific, and struggle with novel spam behaviours, limiting their scalability [3].

2.3 Deep Learning and Hybrid Approaches

To overcome the limitations of purely content-based methods, hybrid approaches integrating textual, behavioural, and network features emerged. DeepBot [4] used RNNs (LSTM/GRU), CNNs, and GNNs combined with account age, posting frequency, sentiment, and network centrality, achieving 90% accuracy and 0.90 F1 score across multiple bot types. Preprocessing steps such as tokenization, stopword removal, stemming, and normalization enhanced performance and reduced noise [4].

Multimodal frameworks like TMTM [5] incorporated user profile features, semantic text embeddings, and graph-based relational structures. Using the TwiBot-22 dataset, the model captured profile-based metrics (e.g., account age, follower–following ratio), textual features from posts and bios, and social network structure, outperforming previous state-of-the-art approaches by 5.48% in detection accuracy [5]. Deep neural network architectures at the tweet-level, such as Contextual LSTM [8], demonstrated that bot detection can achieve high accuracy using minimal features, combining textual content and metadata (e.g., hashtags, retweet count) with robust handling of imbalanced datasets.

Other hybrid approaches, like HHD^{2}SCNN [7], leveraged user profile and tweet-level features, combined with advanced feature selection (Binary Golden Search Optimization), to achieve 98.4% accuracy, demonstrating the potential of optimized hybrid models in real-world scenarios. While these methods are powerful, they remain computationally intensive and reliant on labelled datasets.

2.4 Unsupervised and Structure-Aware Approaches

Scalability and detection of zero-day bot behaviours are primary reasons to use unsupervised, structure-aware methods. UnDBOT [11] uses structural information theory on multi-relational social graphs to partition users into communities and identify bot clusters via influence and cohesion metrics. BotDCGC uses Graph attentional autoencoders with contrastive learning [12] generate node embeddings that separate human and bot behaviours without labels, enhancing adaptability to evolving networks. MulBot [13] leverages LSTM autoencoders to compress multivariate time series of user activity, capturing temporal patterns and detecting new bot types across multiple timelines.

Network-centric approaches focus on coordinated behavior. For instance, co-commenter graphs constructed on YouTube revealed “commenter mobs” that manipulate engagement metrics and propagate misinformation [15,16]. Graph2Vec embeddings with UMAP and K-means clustering uncovered highly coordinated groups, while structural network features such as degree, density, modularity, and clique size provided interpretable insights [15,16]. Online adaptive anomaly thresholding [14] further improves unsupervised detection by dynamically adjusting thresholds to evolving user activity, maintaining consistent false positive and negative rates.

2.5 Synthesis and Research Gap

Overall, research shows a clear progression: supervised content-based methods [1,3] provide high accuracy but low adaptability; hybrid deep learning approaches [4,5,7,8] improve robustness but remain computationally intensive and dataset-dependent; unsupervised, structure-aware frameworks [11–16] offer interpretability and adaptability. Despite these advances, a gap remains for a scalable, interpretable, and unsupervised framework tailored specifically for YouTube’s commenting ecosystem, capable of detecting both individual and coordinated bots. PROS (Pivot and Seek Rank-One Submatrix) addresses this gap by modelling normal user behaviour and detecting anomalies using structural, temporal, and behavioural features, offering a transparent and scalable solution for emerging automated activity.

3 Proposal

3.0.1 Novelty. The core novelty of this research is the first application of the PROS anomaly detection methodology to the YouTube commenting ecosystem. This is a critical domain shift from the original paper’s use of Twitter.

We enhance the model by extracting unique, platform-specific features such as Channel Maturity (channelDate, subscriber, and video counts) and Engagement Context (comment time and like counts). This allows PROS to effectively create categorical bins that distinguish new, high-velocity bot profiles from stable, legitimate user behaviour, resulting in a more interpretable and scalable bot detection framework.

3.0.2 Challenge Level: Weight. The project’s High challenge level (Weight) stems from adapting the novel, unsupervised PROS methodology to the adversarial YouTube commenting ecosystem. The primary hurdle is methodological (50%), requiring the construction of a non-standard hybrid ground truth to reliably evaluate the detection of zero-day bot activity. Technically (30%), success hinges on the robust feature engineering of continuous metrics into high-quality categorical bins and the successful validation of conditional independence between Pivot and Seek features, which is highly complex in correlated social media data. Finally (20%), the engineering challenge demands the system be scalable and highly performant for near real-time processing of millions of records, ensuring the PROS output translates swiftly into actionable policy rules capable of countering fast-moving "commenter mobs."

3.1 Proposed System / Prototype Design

3.1.1 Project Objectives. The goal of this project is to apply the PROS methodology for detecting automated (bot) traffic using unsupervised learning. [17] The first step is to identify categorical web features that are compatible with the PROS framework. Using these features, we will locate feature bins least affected by automation which are, regions of data likely to represent clean, human traffic and use them to construct a baseline clean distribution. This clean model will then be compared against the overall traffic distribution to detect deviations indicative of automated activity. To evaluate performance, we will compare PROS with an alternative unsupervised approach, such as Isolation Forest. Finally, we will derive interpretable rules that highlight which features or feature combinations are most strongly associated with bot-like behaviour.

3.1.2 What is New. The PROS method introduces a novel unsupervised approach that directly handles categorical data in highly noisy environments. Traditional unsupervised models typically require transforming categorical features into numerical ones using techniques like one-hot encoding. However, one-hot encoding often performs poorly in such contexts because categorical variables lack a natural notion of distance between categories. PROS overcomes this limitation by avoiding one-hot encoding altogether. Instead, it identifies "buckets" of traffic that exhibit little to no malicious activity under conditional independence assumptions between certain features. This allows the model to accurately estimate the clean distribution of benign traffic without needing numerical feature representations.

3.1.3 Methodology of Prototype. We begin by modelling the observed web traffic as a mixture of benign and automated (bot) requests. Let C denote the set of clean samples and B the set of bot samples. The total observed traffic is therefore $|C| + |B|$, and we define the fraction of clean traffic as:

$$\alpha = \frac{|C|}{|C| + |B|}. \tag{1}$$

Each request is represented by a feature vector $x = (x_0, x_1, \ldots, x_{M-1})$, where each x_i corresponds to a categorical web feature (e.g., browser version, country, or hour of day). The parameter $\alpha \in (0, 1]$ captures the proportion of benign traffic, while $1 - \alpha$ represents the proportion of bot traffic in the dataset. The observed feature distribution is a mixture of the clean and bot distributions:

$$P(x) = \alpha\, P(x \mid \text{benign}) + (1 - \alpha)\, P(x \mid \text{bot}). \tag{2}$$

Here $P(x \mid \text{benign})$ and $P(x \mid \text{bot})$ are the feature distributions conditioned on clean and automated traffic respectively.

In practice, bot traffic may originate from multiple independent attackers. Let Q denote the number of distinct bot sources, with mixture weights w_q such that $\sum_{q=0}^{Q-1} w_q = 1$. Then:

$$P(x \mid \text{bot}) = \sum_{q=0}^{Q-1} w_q\, P(x \mid \text{bot}_q). \tag{3}$$

When $Q = 1$, the attacker's behaviour may be treated as independent across features; however, for $Q > 1$, the mixture of multiple behaviours breaks this independence. In contrast, we assume that benign traffic features are approximately independent:

$$P(x \mid \text{benign}) = \prod_i P(x_i \mid \text{benign}). \tag{4}$$

This means that, for example, browser version does not depend on time of day, and so on. This gives that the overall observed distribution is:

$$P(x) = \alpha \prod_i P(x_i \mid \text{benign}) + (1 - \alpha) P(x \mid \text{bot}), \tag{5}$$

and the marginal observed distribution along a single feature x_j is:

$$P(x_j) = \alpha\, P(x_j \mid \text{benign}) + (1 - \alpha)\, P(x_j \mid \text{bot}). \tag{6}$$

Consider restricting attention to a single feature bucket like country type $x_k = x'_k$. Within that bucket, the distribution of another feature x_j can be written as

$$P(x_j \mid x_k = x'_k) = \alpha'_k\, P(x_j \mid \text{benign}, x'_k) + (1 - \alpha'_k)\, P(x_j \mid \text{bot}, x'_k), \tag{7}$$

where the local clean fraction is:

$$\alpha'_k = \frac{|C(x_k = x'_k)|}{|C(x_k = x'_k)| + |B(x_k = x'_k)|}. \tag{8}$$

If a bucket $x_k = x'_k$ receives no attack traffic ($P(x_j \mid \text{bot}, x'_k) = 0$), then $\alpha'_k = 1$ and

$$P(x_j \mid x'_k) = P(x_j \mid \text{benign}). \tag{9}$$

Thus, the clean distribution of any feature x_j can be estimated from buckets of another feature that are known to be clean. This is the key insight that allows PROS to recover clean distributions without labelled data. Given the clean model, the relative odds that an observation x is malicious are:

$$\frac{P(\text{bot} \mid x)}{P(\text{benign} \mid x)} = \frac{P(x) - \alpha \prod_i P(x_i \mid \text{benign})}{\alpha \prod_i P(x_i \mid \text{benign})}. \tag{10}$$

The numerator measures how much more frequently x occurs than predicted by the clean model, while the denominator normalizes by the expected benign probability. A larger ratio implies a higher likelihood of automation. This would leave α as the only unknown value. Fortunate for us,the ordering of suspiciousness scores does not depend on α. To see this, denote the left-hand side of Equation (9) as $\text{Odds}(x, \alpha)$. If $\text{Odds}(x, \alpha) > \text{Odds}(y, \alpha)$, then:

$$\frac{P(x)}{\alpha\, P(x \mid \text{benign})} - 1 > \frac{P(y)}{\alpha\, P(y \mid \text{benign})} - 1, \tag{11}$$

which simplifies to

$$P(x)\, P(y \mid \text{benign}) > P(y)\, P(x \mid \text{benign}). \tag{12}$$

This shows that the ordering of suspiciousness does not depend on the choice of α.

The assumption that all features are independent is idealistic and not realistic; in practice, many features will have correlations. For example, Firefox might represent the market in France, but only (25%) in Mexico,indicating a dependence between browser family and country. To address these dependencies, the PROS methodology divides the feature space into smaller subsets where conditional independence holds a higher chance. Let the dataset be partitioned into subsets S_i such that features within each subset are approximately conditionally independent. Each subset might correspond to, for example, a specific country, browser family, or combination of both. Within a subset S_i, the observed distribution can be expressed as:

$$P(\mathbf{x} \mid S_i) = \alpha_i P(\mathbf{x} \mid \text{benign}, S_i) + (1 - \alpha_i) P(\mathbf{x} \mid \text{bot}, S_i), \tag{13}$$

where α_i is the fraction of benign traffic within subset S_i.

For a particular feature x_j within subset S_i,

$$P(x_j \mid S_i) = \alpha_i P(x_j \mid \text{benign}, S_i) + (1 - \alpha_i) P(x_j \mid \text{bot}, S_i). \tag{14}$$

This mirrors the equation (2) but applies locally to each subset.

Suppose there exists a set of untainted feature values $\mu(\lambda(x_j))$ for which no bot traffic occurs. Then the observed distribution within those buckets is equal to the true clean distribution:

$$P(x_j \mid S_i, \mu(\lambda(x_j))) = P(x_j \mid \text{benign}, S_i). \tag{15}$$

Thus, by identifying untainted buckets, we can recover the benign distribution for each feature without labeled data. After estimating the clean distribution, the odds that an observation is malicious can be expressed as:

$$\frac{P(\text{bot} \mid \mathbf{x}, S_i)}{P(\text{benign} \mid \mathbf{x}, S_i)} = \frac{P(\mathbf{x} \mid S_i)}{\alpha_i \prod_j P(x_j \mid \text{benign}, S_i)} - 1. \tag{16}$$

The numerator measures how much more frequently a feature combination occurs than predicted by the clean model, while the denominator normalizes by the expected benign probability. A larger ratio indicates higher bot likelihood. If multiple buckets (e.g., “Chrome71”, “Iowa”, “Oregon”) exhibit identical or highly similar marginal distributions, they are likely untainted. Formally, for buckets b_m and b_n it is expressed as:

$$P(x_j \mid S_i, b_m) = P(x_j \mid S_i, b_n) = P(x_j \mid \text{benign}, S_i). \tag{17}$$

Approximate equality can be verified using divergence metrics such as Kullback–Leibler (KL) divergence to measure statistical distance. The relationship between the observed, clean, and bot distributions can be written in matrix form:

$$\Omega = \alpha\Sigma + (1-\alpha)\Theta, \tag{18}$$

where observed distributions is Ω, clean (benign) rank-1 matrix is Σ and bot distribution matrix is Θ

Clean traffic produces a low-rank (rank-1) structure, whereas bot traffic introduces additional dimensions (higher rank). Hence, identifying clean distributions corresponds to finding rank-1 submatrices within Ω.

3.1.4 Algorithm 1: Finding Clean Distributions.

(1) For each subset S_i and feature x_j, pivot the data into a matrix $\Omega(x_j)$.
(2) Identify the largest cluster of co-linear columns in $\Omega(x_j)$, corresponding to untainted (clean) buckets.
(3) Estimate the clean distribution for each feature as $\hat{P}(x_j \mid \text{benign}, S_i)$ by averaging over the untainted buckets.

3.1.5 Algorithm 2: Calculating Bot Odds. Once clean distributions are estimated, the odds of automation for any observation can be computed as:

(1) For each subset $S_i \in \{S_i\}$:

$$\text{Odds}(x \mid S_i) = \frac{P(x \mid S_i)}{0.5 \prod_j \hat{P}(x_j \mid \text{benign}, S_i)} - 1. \tag{19}$$

(2) The computed odds quantify how much more frequently a feature combination occurs than predicted by the clean model. Higher odds indicate greater likelihood of automated traffic.

3.1.6 Who cares.

(1) Strategic Policy Actionability - Many unsupervised and deep learning approaches, while effective, are black boxes [11, 12]. They provide an anomaly score but no transparent, human-readable reason for the classification. This creates a critical bottleneck for security analysts who must manually audit the results and convert them into platform policy. The core power of the PROS framework (RQ3) lies in its interpretable output: the Pivot and Seek rules. These rules directly identify the conditional feature combinations responsible for the anomaly (e.g., "Accounts created within the last 24 hours AND posting frequency > 10 comments/min"). This transforms detection into actionable policy, allowing platform administrators to automate enforcement instantly and transparently, dramatically reducing the time and cost associated with manual analyst investigation.
(2) Domain-Specific Threat Mitigation - This project specifically targets Youtube’s unique threat profile, which extends beyond simple content spam:
    (a) Organized Coordination: We move past individual account analysis to uncover the structural, coordinated activity of "commenter mobs" [15, 16] that manipulate engagement metrics and skew viral narratives.
    (b) Financial and Youth Protection: By validating our model against structural features associated with scam campaigns [2] (like unusual velocity metrics), PROS protects vulnerable user populations from high-impact threats such as romance and voucher scams that proliferate in the comments section.

3.2 Proposed Evaluation Method

The proposed evaluation method is designed to rigorously quantify the effectiveness, efficiency, and interpretability of the PROS methodology in detecting automated commenting behaviour on Youtube. This evaluation serves as the "final exam" for project success and addresses the novelty and challenge level inherent in applying an unsupervised method to a dynamic, noisy and high-volume platform.

3.2.1 Mid-Term Success. By the mid-term stage, success is defined by the successful implementation and validation of the PROS core algorithm. The algorithm should be able to take a dataset as a parameter and establish the optimal set of categorical and structural features (RQ2), whilst also demonstrating the algorithm’s ability to identify a set of "unattacked bins" to generate a preliminary, clean distribution of benign traffic on a significant sample of the Youtube dataset.

3.2.2 Final Success. The final project success is quantified by performance metrics and the achievement of interpretability goals:

(1) Effectiveness: The PROS classifier must demonstrate superior performance compared to established unsupervised baselines, with a focus on maximising Recall to minimize false negatives (undetected bots) and maintaining a competitive F1-Score.
(2) Interpretability: The resulting decision rules (the identified safe feature bins) must provide transparent, human-readable insights into novel bot strategies. (RQ3)

3.2.3 Quantitative Performance Metrics. To address the challenge of evaluating an unsupervised model, we will establish a hybrid ground truth using 2 approaches: (1) - existing accounts flagged by Youtube’s public API or content filtering rules and (2) a small, manually-labeled set of comments exhibiting clear signs of coordinated or mass-posting behaviour (such as identical, non-contextual, or hyper-frequent) comments identified during data collection.

The primary, quantitative metrics for evaluating the anomaly scoring performance will be:

(1) Area Under the ROC Curve (AUC): Since PROS outputs an anomaly score, AUC will be the core metric to assess the model's overall discriminative power across all possible threshold settings. A higher AUC indicates better separation between benign and automated traffic.

(2) F1-Score, Precision, and Recall: These will be calculated by setting a pragmatic anomaly threshold on the PROS score distribution. Recall will be prioritized, as failure to detect new bot strategies is the most costly outcome for platform integrity. Precision will ensure that the rules provided to human analysts are reliable and minimize manual false-positive investigations.

(3) Baseline Comparison: The full PROS framework will be rigorously benchmarked against the Isolation Forest (IF) algorithm, run on the same, one-hot encoded feature matrix. The evaluation will demonstrate that PROS is less susceptible to noise and better suited for the high dimensionality of categorical data compared to traditional distance-based or tree-based unsupervised methods.

3.2.4 *Scalability and Interpretability Evaluation.*

(1) Scalability: We will measure the training and inference time complexity of both PROS and the baseline model on incrementally scaled dataset volumes (e.g., 10^5, 10^6, 10^7 records). This confirms PROS's practical viability for large-scale, real-time deployment, which is a key requirement of the 'who cares' objective.

(2) Interpretability and Novelty: This qualitative evaluation will address the core novelty of the project (RQ3). We will analyze the resulting Pivot and Seek decision rules to:
   (a) Feature Correlation: Identify which combinations of structural and behavioural features are conditionally independent, and which are highly correlated with bot activity.
   (b) Rule Actionability: Present the top 10 generated human-readable rules (e.g., "Accounts with age < 30 days AND posting frequency > 5 posts/minute are highly anomalous") to demonstrate the interpretability and direct actionability for platform policy enforcement.

4 Project Progress to Date

4.1 Materialised Progress

The biggest step in our implementation is collection and gathering vast amounts of metadata regarding Youtube comments. This meant that most of our progress was about scraping as much data as possible, to enable our PROS technique to be as efficient as possible.

4.1.1 Project Scope Refinement. After creating the initial Python scraping script and reviewing the detailed metadata fields exposed by the YouTube Data API, we were able to strategically refine our data collection scope. We confirmed the availability of crucial features, such as account creation dates and velocity metrics, which are essential for the PROS technique. This technical review led us to focus our collection efforts on three distinct and high-volume YouTube channel categories: gaming, music, and politics. We decided to use the PROS technique on these categories, so that we can understand the proportion of bot comments on each Youtube category (RQ4) and also highlight whether PROS performs better than other algorithms such as Isolation Forests on specific channels.

4.1.2 Data Collection. We created a Google Cloud project to utilize the YouTube Data API, which enables us to systematically extract comprehensive comment metadata, including channel identifiers, account creation dates, subscriber counts, and textual content. We developed a Python script which scrapes the data into a JSON-formatted file using the API. This process has successfully ingested metadata for over 300,000 comments, along with their associated channel data, covering the selected gaming, music, and politics categories. The key variables successfully extracted from the API and stored include:

(1) Channel Profile Metrics: channelDate (account creation date), channelSubscriberCount, channelVideoCount, and channelViewCount.

(2) Comment Engagement Metrics: commentDate (published time), commentLikeCount, and the commentText itself.

4.1.3 Project Plan. Since our data collection process is the most vital stage, our main focus is on establishing a robust, automated data pipeline to continuously retrieve and store comment metadata from the YouTube Data API. All scraped data will be ingested into a Pandas DataFrame for centralized, high-speed manipulation. A major focus will be on Feature Engineering (RQ2), where continuous variables (e.g., comment timestamp, account age) are transformed and discretized into the high-quality categorical bins required by the PROS algorithm. Key derived features like Channel Maturity (aggregating account age, subscription, and video counts) will be calculated and validated for consistency across the dataset.

4.2 Group Dynamics

4.2.1 *a. division of labour statement (who did what).*

(1) Rushabh Dharamshi: Researched relevant literature papers on bot detection and took part in writing the related work and proposal. Helped in scraping and gathering data using Youtube Data API by developing and utilising a Python script.
(2) Osama Heweitat: Researched relevant literature papers on bot detection and took part in writing the related work and proposal. Helped in scraping and gathering data using Youtube Data API by developing and utilising a Python script.
(3) Radoslaw Kucisz: Researched relevant literature papers on bot detection and took part in writing the related work and proposal. Helped in scraping and gathering data using Youtube Data API by developing and utilising a Python script.
(4) Bharatraj KR: Researched relevant literature papers on bot detection and took part in writing the related work and proposal. Helped in scraping and gathering data using Youtube Data API by developing and utilising a Python script.

4.2.2 *individual contribution statement (%) .*

(1) Rushabh Dharamshi (25%)
(2) Osama Heweitat (25%)
(3) Radoslaw Kucisz (25%)
(4) Bharatraj KR (25%)

References

[1] Jovian Yanto, Tandiono, R.D., Wulandari, L.A., and Ghinaa Zain Nabiilah. (2025). *Spam Detection on YouTube Comment Section: Comparison Between Deep Learning and Machine Learning Methods.* pp. 753–759. https://doi.org/10.1109/IAICT65714.2025.11100517.

[2] Seung Ho Na, Cho, S., and Shin, S. (2023). *Evolving Bots: The New Generation of Comment Bots and their Underlying Scam Campaigns in YouTube.* https://doi.org/10.1145/3618257.3624822.

[3] Aiyar, S., and Shetty, N.P. (2018). *N-Gram Assisted Youtube Spam Comment Detection. Procedia Computer Science*, 132, pp. 174–182. https://doi.org/10.1016/j.procs.2018.05.181.

[4] Alnahari, M. (2025). *DeepBot: Bot Detection in Twitter Using Deep Learning Method for Social Media Security. Communications in Computer and Information Science*, pp. 308–323. https://doi.org/10.1007/978-3-032-01948-6_20.

[5] Olmar Arranz-Escudero, Quijano-Sanchez, L., and Liberatore, F. (2025). *Enhancing misinformation countermeasures: a multimodal approach to twitter bot detection. Social Network Analysis and Mining*, 15(1). https://doi.org/10.1007/s13278-025-01435-w.

[6] Aljabri, M., Zagrouba, R., Shaahid, A., Alnasser, F., Saleh, A., and Alomari, D.M. (2023). *Machine learning-based social media bot detection: a comprehensive literature review. Social Network Analysis and Mining*, 13(1). https://doi.org/10.1007/s13278-022-01020-5.

[7] Kumar, A.V.S., Kumar, N.S., and Devi, R.K. (2025). *An efficient hybrid Hopfield convolutional neural network for detecting spam bots in Twitter platform. International Journal of Cognitive Computing in Engineering*, 6, pp. 569–587. https://doi.org/10.1016/j.ijcce.2025.05.003.

[8] Kudugunta, S., and Ferrara, E. (2018). *Deep neural networks for bot detection. Information Sciences*, 467, pp. 312–322. https://doi.org/10.1016/j.ins.2018.08.019.

[9] Yang, K., Varol, O., Davis, C.A., Ferrara, E., Flammini, A., and Menczer, F. (2019). *Arming the public with artificial intelligence to counter social bots. Human Behavior and Emerging Technologies*, 1(1), pp. 48–61. https://doi.org/10.1002/hbe2.115.

[10] Cresci, S. (2020). *A decade of social bot detection. Communications of the ACM*, 63(10), pp. 72–83. https://doi.org/10.1145/3409116.

[11] Peng, H., Zhang, J., Huang, X., Hao, Z., Li, A., Yu, Z., and Yu, P.S. (2024). *Unsupervised Social Bot Detection via Structural Information Theory. ACM Transactions on Office Information Systems.* https://doi.org/10.1145/3660522.

[12] Wang, X., Wang, K., Chen, K., Wang, Z., and Zheng, K. (2024). *Unsupervised twitter social bot detection using deep contrastive graph clustering. Knowledge-Based Systems*, 293, p. 111690. https://doi.org/10.1016/j.knosys.2024.111690.

[13] Mannocci, L., Cresci, S., Monreale, A., Vakali, A., and Tesconi, M. (2022). *MulBot: Unsupervised Bot Detection Based on Multivariate Time Series. arXiv.org.* https://arxiv.org/abs/2209.10361. [Accessed 6 Nov. 2025].

[14] Sun, S., Sankararaman, A., and Narayanaswamy, B. (n.d.). *Online Adaptive Anomaly Thresholding with Confidence Sequences.* [online]. Available at: https://assets.amazon.science/79/8f/8f8cbd824adf9229d6d1248f2ca1/online-adaptive-anomaly-thresholding-with-confidence-sequences.pdf [Accessed 6 Nov. 2025].

[15] Shajari, S., and Agarwal, N. (2025). *Developing a network-centric approach for anomalous behavior detection on YouTube. Social Network Analysis and Mining*, 15(1). https://doi.org/10.1007/s13278-025-01417-y.

[16] Shajari, S., Agarwal, N., and Alassad, M. (2023). *Commenter Behavior Characterization on YouTube Channels. arXiv.org.* Available at: https://arxiv.org/abs/2304.07681.

[17] Herley, C. (n.d.). *Automated Detection of Automated Traffic.* Available online: https://www.usenix.org/system/files/sec22-herley.pdf [Accessed 6 November 2025].

5 appendix

Symbol	**Description**
x_k	Categorical feature
N_k	Cardinality of x_k
$\mathbf{x}$	Vector of features, e.g., $(x_0, x_1, x_2, \ldots, x_{M-1})$
S_i	Subset of the data; e.g., a single country
$P(\mathbf{x})$	Observed distribution of $\mathbf{x}$
$P(\mathbf{x} \mid \text{benign})$	Distribution of $\mathbf{x}$ in benign traffic
$P(\mathbf{x} \mid \text{bot})$	Distribution of $\mathbf{x}$ in malicious traffic
$\hat{P}(\mathbf{x} \mid \text{benign})$	Estimate of $P(\mathbf{x} \mid \text{benign})$
α	Fraction of traffic that is benign
$\lambda(x_j)$	Set of features conditionally independent of x_j given S_i
$\mu(\lambda(x_j))$	Set of unattacked buckets of features in $\lambda(x_j)$

Table 1: Explanation of variables used.