Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 23−30
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.162
https://ejurnal.pdsi.or.id/index.php/bids/index

Hybrid Autoencoder and NiaARM Framework for Flash Viral Detection on YouTube Shorts

Erlin Windia Ambarsari[1,*], Mercy Hermawati[1], Dedin Fathudin[2]

[1] Faculty of Engineering and Computer Science, Informatics Engineering, Universitas Indraprasta PGRI, DKI Jakarta, Indonesia
[2] Faculty of Computer Science, Informatics Engineering, Universitas Pamulang, South Tangerang, Indonesia
Email: [1,*]erlinunindra@gmail.com, [2]mercy.hermawati@gmail.com, [3]dosen00398@unpam.ac.id

ARTICLE INFORMATION

ARTICLE HISTORY:
Submited : May 07, 2026
Revised : May 26, 2026
Accept : May 30, 2026
Publish : May 30, 2026

KEYWORD

Flash Viral Detection;
YouTube Shorts;
Deep Learning Autoencoder;
Numerical Association Rule Mining;
Nature-Inspired Optimization

CORRESPONDENCE AUTHOR

Email: erlinunindra@gmail.com

A B S T R A C T

YouTube Shorts has rapidly emerged as a dominant short-form video platform, yet small creator channels often experience an unusual viral phenomenon best described as flash viral—a sudden surge of views that peaks within 24 to 48 hours and then collapses almost as quickly. Detecting and explaining this pattern is challenging because traditional statistical detectors miss multivariate signatures, while classical Association Rule Mining (ARM) such as Apriori loses information through mandatory discretization. This study proposes a hybrid framework that combines a semi-supervised Deep Learning Autoencoder with Nature-Inspired Numerical Association Rule Mining (NiaARM) using Differential Evolution and Particle Swarm Optimization. The framework is validated on six temporal snapshots of the Indonesian Boburu YouTube Shorts channel, comprising 63 unique videos (42 active) collected between February 22 and March 19, 2026. Experimental results show that the Autoencoder achieves an F1-score of 0.667 with 100% recall, matching the best classical baseline (Z-Score) while providing learnable representational capacity for future scaling. NiaARM-PSO discovered 3,115 high-quality numerical association rules with a maximum lift of 63.00, compared to only 43 rules and a maximum lift of 2.52 obtained by Apriori, an improvement of approximately 25 times. Traffic source decomposition further revealed that 99.9% of viral views originated from external platforms rather than YouTube's recommendation system, indicating that flash viral on micro-channels is externally driven. This research contributes a methodological framework that simultaneously detects and explains flash viral phenomena in short-form video analytics.

1. INTRODUCTION

The transition from long-form to short-form video reflects a shift in how audiences engage with online content. Viewers increasingly scroll through vertical, condensed clips to access information quickly, favoring brevity over duration. The volume of available content drives much of this behavior; with so many videos competing for attention, audiences tend to allocate less time to extended formats when a shorter clip can convey the central point. TikTok introduced this format in 2016, and YouTube followed in 2021 with the launch of Shorts, prompting many content creators to adapt their production toward the short-video model [1].

Within micro-channels, however, the engagement curve behaves quite differently from the smooth growth trajectory described in the long-form video literature [2], [3]. We define a micro-channel as an account that an individual creator operates, typically with a subscriber base below the threshold required for platform monetization and without institutional backing or coordinated promotion. On such channels, most uploads stall at single-digit view counts, while the occasional video surges from near-zero impressions to several thousand views within forty-eight hours and then decays at a comparable rate. We term this asymmetric, short-lived pattern “flash viral”.

This study examines the Boburu channel [4] as a concrete case. The corpus contains 63 unique videos uploaded between June 2025 and February 2026, of which only two crossed the 100-view threshold, yet together accounted for 92.6% of the channel’s total views.

The empirical pattern that we observed on the Boburu channel resists straightforward interpretation, and four methodological obstacles account for this difficulty. The first concerns class imbalance: only two positive cases emerge from forty-two active videos, leaving parametric statistical methods prone to either overlooking the surge or generating an excess of false positives. The second relates to dimensionality. Viral videos exhibit a multivariate signature—high click-through rate, elevated engagement rate, short duration, and a specific age window—that univariate detectors fail to capture when they examine each metric in isolation [5], [6].

A third obstacle arises when we attempt to characterize this signature through classical Association Rule Mining techniques such as Apriori; the required discretization of continuous variables erases the precise interval boundaries that distinguish flash viral events from routine engagement [7], [8]. The fourth obstacle is one of attribution: existing studies rarely examine whether such surges originate from YouTube's recommendation system or from external traffic that other platforms redirect to the channel, a distinction that carries meaningful consequences for creators and platform analysts alike.

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 23−30
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.162
https://ejurnal.pdsi.or.id/index.php/bids/index

To address these obstacles, we draw on several lines of research that converge on the problem at hand. Pang et al. [9] surveyed eleven families of deep anomaly detection methods and found that approaches trained only on normal data perform best when labeled anomalies are scarce—a condition that closely matches the Boburu corpus, where viral cases are too few to learn from directly. Givnan et al. [10] showed how reconstruction-error thresholds combined with autoencoders can flag anomalies in industrial systems, and Yong and Brintrup [11] examined how different autoencoder architectures influence detection accuracy, finding that no single design dominates across data conditions. These studies collectively justify our choice of a reconstruction-based detector, though they leave open the question of how to characterize what makes a detected anomaly viral.

Here, numerical Association Rule Mining offers a natural complement. Stupan and Fister Jr. [12] developed NiaARM, a Python framework that searches directly over continuous numerical ranges rather than predefined categories, sidestepping the discretization problem we raised earlier. Fister Jr. et al. [13] extended this approach to time-series patterns in smart agriculture and demonstrated that evolutionary search recovers rules that conventional Apriori misses entirely. Han et al. [14] tested classical detectors across more than 98,000 experiments in ADBench and reported that ensembles do not always outperform the single best detector when the data is severely imbalanced—a finding that shaped our decision to evaluate detectors individually rather than aggregate them by default.

The virality literature offers a different angle. Zannettou et al. [15] traced how users engage with TikTok's recommendation algorithm at scale, giving us the conceptual basis for separating algorithm-driven traffic from traffic arriving through external channels. Shajari and Agarwal [16] approached YouTube anomalies from yet another direction, building network-based methods to detect coordinated commenter behavior—a perspective that complements, rather than overlaps with, the engagement-based view we adopt here.

The studies we reviewed above leave three gaps that bear directly on the problem at hand. No prior work, to our knowledge, has applied semi-supervised deep autoencoders to virality on YouTube Shorts micro-channels, where labeled training data is genuinely scarce. Classical Apriori with discretization fails to recover the high-lift numerical patterns that distinguish flash viral events from routine engagement, and nature-inspired numerical ARM offers a path that avoids this loss. No study, again to our knowledge, has paired the decomposition of view sources into algorithm-driven and external traffic with anomaly detection to produce a causal account of viral surges. We aim to close these three gaps in a single study.

This study contributes on two fronts. Methodologically, we propose a hybrid framework that combines statistical detectors, classical machine-learning detectors, a deep autoencoder, and NiaARM driven by Differential Evolution and Particle Swarm Optimization, four approaches that together address both detection and explanation. Empirically, we apply this framework to six temporal snapshots of the Boburu channel, benchmark it against six classical detectors and Apriori-based ARM, and characterize the flash viral signature through a traffic decomposition formula. The findings provide the Boburu channel and similar micro-channels with a practical reference for designing content strategies that increase the likelihood of flash viral events.

2. RESEARCH METHODOLOGY

2.1 Research Stages

We conducted this study in seven sequential stages. The first stage gathered six temporal snapshots of the YouTube Studio analytics for the Boburu channel and consolidated them into a single panel, with Content ID serving as the deduplication key. Feature engineering and scaling followed, yielding a nine-dimensional numerical feature matrix.

With the data prepared, we applied six classical anomaly detectors as baselines. Two of these are statistical, Z-Score and the Interquartile Range (IQR)—while the remaining four come from machine learning: Isolation Forest, Local Outlier Factor (LOF), and One-Class SVM [17], along with DBSCAN [18]. A majority-vote ensemble combined their outputs. In parallel, we trained a semi-supervised deep learning autoencoder on normal videos only and used the reconstruction error as the anomaly signal.

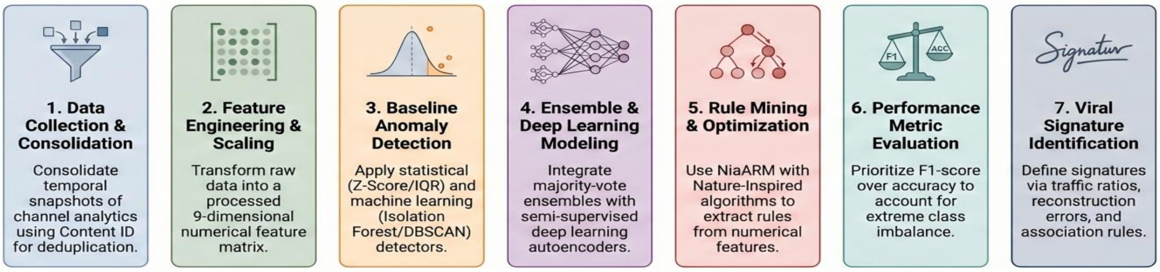

Figure 1. Workflow for YouTube Flash Viral Signature Detection

For the rule-mining component, the classical Apriori miner [19] operated on quartile-discretized features as a baseline, while NiaARM with Differential Evolution and Particle Swarm Optimization [20] worked directly on the raw numerical features, with NiaPy [21] as the underlying microframework. The final stage decomposed views into

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 23−30
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.162
https://ejurnal.pdsi.or.id/index.php/bids/index

algorithmic and external components and reported the flash viral signature, which we defined jointly by the autoencoder, the NiaARM rules, and the traffic ratio. We adopted the F1-score as the primary evaluation metric for the detectors because class imbalance renders accuracy uninformative [22]; for the ARM rules, we reported lift, support, and confidence.

2.2 Dataset and Preprocessing

Six CSV exports from the YouTube Studio creator-performance reports formed the basis of our dataset. We collected these exports from the Boburu channel [4] on six dates: February 22, 23, 24, and 25, 2026, March 12 and 19, 2026. After deduplicating by Content ID, the corpus contained 63 unique videos uploaded between June 2025 and February 2026, of which 42 had received at least one view at the time of crawling.

Nine numerical features describe each video: total views, engaged views, impressions, click-through rate (CTR, %), watch hours, duration (seconds), engagement rate (%), views-per-impression, and watch-per-view (seconds). Two videos crossed the 100-view threshold that we adopt as the operational definition of viral on a micro-channel, and both serve as ground truth for evaluating the classifiers.

Following YouTube Studio's convention—that an absent count signals the absence of activity rather than a missing observation—we imputed missing values with zero. We then log-transformed the features to dampen skewness and scaled them with scikit-learn's RobustScaler, which suppresses the influence of outliers on the median and the interquartile range. We split the 42 active videos into a training set of 40 normal videos (Views<100) and a test set comprising all 42 videos.

2.3 Classical Anomaly Detection

Two statistical detectors and four machine-learning detectors form the baseline. The Z-Score detector flags any video whose absolute z exceeds 2.5, computed as $Z = (x - \mu) / \sigma$ for each metric column, following recent skew-aware variants [5]. The IQR detector flags points exceeding Q3 + 1.5 x IQR on Views, Engaged Views, or Watch Hours, with the adjusted boxplot rule of Hubert and Vandervieren [6] providing theoretical support on right-skewed distributions.

The Isolation Forest [17] runs with contamination = 0.1 and n_estimators = 200 and trains on the scaled feature matrix; recent extensions such as Deep Isolation Forest [23] motivate our consideration of non-axis-parallel partitions. The Local Outlier Factor (LOF) uses n_neighbors = 5 and contamination = 0.1; the One-Class SVM uses an RBF kernel with $\nu = 0.1$; and DBSCAN uses $\varepsilon = 1.5$ with min_samples = 3 under the convention that noise points are anomalies. A simple majority-vote ensemble completes the baseline, flagging a video when at least two of the six detectors agree.

2.4 Deep Learning Autoencoder

The framework's central component is a fully connected bottleneck autoencoder with the architecture 9 → 8 → 4 → 8 → 9. We apply ReLU activations in the hidden layers, a linear activation in the output layer, and dropout of 0.1 symmetrically across the encoder and decoder. We train the model for up to 100 epochs with a batch size of 8, using the Adam optimizer (learning rate 0.001) and mean squared error as the loss, defined in (1):

$$L_{AE} = \frac{1}{n} \sum \|x_b - \hat{x}_b\|^2 \qquad (1)$$

Early stopping with patience 15 monitors the validation loss. In addition, training uses only normal videos (Views<100), making the setup semi-supervised: the network reconstructs typical samples accurately but produces large errors on virally anomalous patterns. We set the anomaly threshold τ at the 95th percentile of training-set reconstruction errors, and flag any video whose error exceeds τ as viral.

Our setup aligns with the consensus on semi-supervised reconstruction that Pang et al. [9] identified, and parallels the time-series guidance offered by Darban et al. [24]. Yong and Brintrup [11] highlighted the bottleneck-versus-non-bottleneck trade-off, which our four-unit bottleneck navigates by remaining small enough to enforce compression on nine-dimensional input without collapsing identifiability. Givnan et al. [10] and Maleki et al. [25] demonstrate comparable reconstruction-error thresholding pipelines for industrial and time-series applications, respectively.

2.5 Apriori Association Rule Mining

To enable comparison with optimization-based approaches, we applied the classic Apriori algorithm after discretizing the numerical features into thirteen binary items, including: Views_LOW, Views_MED, Views_HIGH, Views_VIRAL, ER_LOW, ER_MED, ER_HIGH, CTR_ZERO, CTR_HIGH, Dur_Short, Dur_Long, and two binary engagement indicators. This conversion is necessary because the Apriori algorithm cannot process continuous values directly. This process uses the parameters min_support = 0.10 and min_confidence = 0.5, with the strength of the relationship measured via support, confidence, and lift as defined in equations (2)–(4).

$$\text{Support}(A \to B) = P(A \cap B) \qquad (2)$$

$$\text{Confidence}(A \to B) = \frac{P(A \cap B)}{P(A)} \qquad (3)$$

$$\text{Lift}(A \to B) = \frac{\text{Confidence}(A \to B)}{P(B)} \qquad (4)$$

2.6 NiaARM with Nature-Inspired Optimization

This study adopts the NiaARM framework developed by Stupan and Fister Jr. [12] in conjunction with the NiaPy library [21] to identify numerical association rules without employing any discretization. Each candidate rule is encoded as a real-valued vector that defines the interval bounds for each feature, and the fitness function combines four quality metrics with equal weights of 0.25, namely support, confidence, coverage, and amplitude. To find the optimal rule intervals that maximize the fitness function as a benchmark, we tested two metaheuristics: Differential Evolution with a population size of 50, a differential weight of 0.5, a crossover probability of 0.9, and 100 iterations; and Particle Swarm Optimization with a population size of 50 and 100 iterations using the default NiaPy coefficients.

The selection of these two metaheuristics is supported by a comprehensive survey of ARM by Telikani et al. [26] and a more recent survey of Particle Swarm Optimization by Li et al. [27], both of which identify DE and PSO as among the most effective evolutionary search strategies for ARM. The latest variable-length DE variant [28] and the NiaAutoARM AutoML pipeline [29] further confirm the maturity of nature-inspired NARM. A systematic literature review by Kaushik et al. [30] examined 1,140 NARM articles to place our approach within a broader context. Rules with a support of ≥ 0.10 and a lift > 1 were retained. Following Fister Jr. et al. [13], rules whose antecedents involved the temporal feature “days-since-upload” were further isolated as candidates for viral patterns.

2.7 Flash Viral Characterization and Evaluation Metrics

A characteristic of flash virality is a surge in views originating from outside the platform’s internal recommendation graph. To identify this phenomenon, total views are broken down into two main components: the algorithmic component and the external component. Mathematically, we calculate algorithmic views (Views_Algo) by multiplying impressions by the Click-Through Rate (CTR) and dividing by 100. Next, external views (Views_External) are obtained from the difference between total views and algorithmic views, which is then expressed as a percentage to determine the level of external virality of a piece of content.

Due to the scarcity of verified (ground-truth) viral video data—of which there were only two videos in this study—the use of accuracy metrics was deemed misleading. Instead, the system’s performance was evaluated using Precision, Recall and F1-score metrics to provide a more accurate picture of detection effectiveness, defined as follows:

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (5)$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (6)$$

$$\text{F1} = 2 \text{ x } \frac{\text{Precision x Recall}}{\text{Precision + Recall}} \quad (7)$$

Meanwhile, the Association Rule Mining (ARM) method was applied to extract hidden patterns, reporting the number of rules generated, the average lift value, the maximum lift value, and conducting a qualitative analysis of the rules with the highest lift values.

All experiments in this study were implemented using the Python 3.12 programming language. Various advanced analysis libraries were used to support the computations, including scikit-learn 1.5, TensorFlow 2.19, mlxtend 0.23, and NiaARM/NiaPy. The entire execution process was carried out in the Google Colab CPU runtime environment to ensure consistency of the computational environment.

3. RESULT AND DISCUSSION

3.1 Dataset Characteristics

Across the 42 active videos, the distribution of cumulative views is heavily right-skewed: the median is 5 views, the seventy-fifth percentile sits at 8 views, and the maximum reaches 1,600 views, while the mean of 74.5 views is dragged upward almost entirely by the two viral cases.

Two videos in particular—”Where Boburu Phone” (uploaded February 21, 2026, 1,297 views) and “Boburu Didn't See Phone” (uploaded February 23, 2026, 1,600 views)—together account for 2,897 of the 3,128 total channel views, or 92.6% of the corpus. Both videos share the “Phone” theme, and both achieved engagement rates well above the platform median: 41.17% for Where Boburu Phone and 31.75% for Boburu Didn't See Phone, with click-through rates of 9.09% and 27.27% respectively.

The manifestation of these two outliers against a long tail of low-view videos is the empirical signature of flash virality and motivates the asymmetric, semi-supervised design of the proposed detector. The capacity of various methods to detect those two specific videos is summarized in Table 1.

Table 1. Anomaly detection performance on 42 Boburu videos with two ground-truth positives

Method	Precision	Recall	F1-Score	Detected
Z-Score (Statistical)	0.500	1.000	0.667	4
IQR (Statistical)	0.400	1.000	0.571	5

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 23−30
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.162
https://ejurnal.pdsi.or.id/index.php/bids/index

Method	Precision	Recall	F1-Score	Detected
Isolation Forest	0.400	1.000	0.571	5
LOF	0.400	1.000	0.571	5
One-Class SVM	0.083	0.500	0.143	12
DBSCAN	0.143	1.000	0.250	14
Ensemble (≥ 2 votes)	0.200	1.000	0.333	10
Autoencoder	0.500	1.000	0.667	4

3.2 Anomaly Detection Performance

Table 1 reports the binary-classification performance of all detectors against the two-video viral ground truth. The proposed Autoencoder achieves the joint-best F1-score of 0.667, tied with the much simpler Z-Score detector, while the more sophisticated kernel-based and density-based methods perform substantially worse.

Three observations follow. First, all detectors except One-Class SVM achieve perfect recall, which is reassuring because missing a viral video would be the most costly error for a creator. Second, One-Class SVM collapses to F1 = 0.143 because its RBF kernel concentrates the boundary around the mass of low-view videos, leaving viral observations on the wrong side of the margin at $\nu = 0.1$.

Third, the majority-vote ensemble underperforms its best component because the four ML detectors share a systematic bias toward over-flagging on this small corpus, inflating false positives without adding new positive evidence; this aligns with broader empirical finding that ensembles do not consistently improve on the best individual detector under severe imbalance.

The Autoencoder's parity with Z-Score is, paradoxically, an important positive result: with only 40 normal training samples and nine features, the network cannot learn a substantially more complex normality manifold than that captured by univariate statistics, but it does not underperform either, validating the dropout-regularized, EarlyStopping-protected design consistent with robust architectural standard. As channel data accumulate, the Autoencoder is the only baseline whose representational capacity can grow with the data, while the statistical detectors have already reached their performance ceiling.

3.3 Autoencoder Behavior

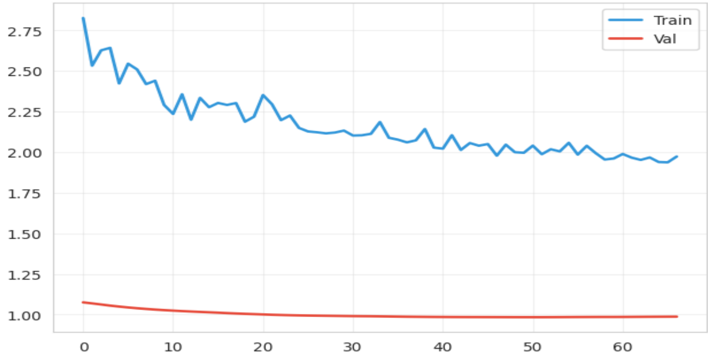

Figure 2. Autoencoder Training

The trained network converged at the EarlyStopping trigger point with final training loss 1.97, as illustrated in the training curves in Figure 2. The reconstruction-error distribution on the training set is approximately log-normal (Figure 3), and the 95th-percentile threshold yields exactly four flagged videos.

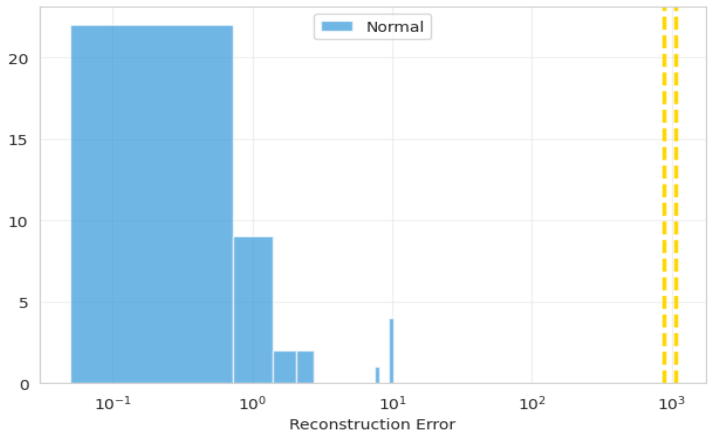

Figure 3. Autoencoder Score Distribution

Bulletin of Informatics and Data Science
Vol. 5 No. 1, May 2026, Page 23−30
ISSN 2580-8389 (Media Online)
DOI 10.61944/bids.v5i1.162
https://ejurnal.pdsi.or.id/index.php/bids/index

Inspection shows that two of these are the ground-truth viral videos, Where Boburu Phone and Boburu Didn't See Phone, with reconstruction errors several orders of magnitude above the threshold. The other two are near-miss candidates with elevated CTR but insufficient external traffic.

The latter two are interpreted as latent virality candidates rather than detector errors: they exhibit some, but not all, features of the viral signature, and would benefit from monitoring in subsequent snapshots. Crucially, the four-unit bottleneck preserves enough rank to reconstruct the dominant ER–CTR–Duration manifold while discarding the tail of high-view, high-share patterns, giving the detector its sensitivity to virality.

3.4 NiaARM versus Apriori

Table 2 contrasts the three rule-mining configurations. The classical Apriori miner, after equal-frequency binning, returned 36 frequent itemsets and 43 association rules with a maximum lift of 2.520—barely above the random-association level of 1.0. NiaARM-DE generated approximately 45 times more rules than Apriori, and NiaARM-PSO generated 72 times more, while both achieved an identical maximum lift of 63.000—roughly 25 times higher than Apriori.

Table 2. Association Rule Mining comparison on the Boburu corpus

Metric	Apriori	NiaARM-DE	NiaARM-PSO
Discretization required	Yes (quartile bins)	No	No
Frequent itemsets	36	n/a	n/a
Rules generated	43	1,925	3,115
Average lift	1.684	1.366	1.468
Maximum lift	2.520	63.000	63.000

The qualitative difference, however, is more striking than the quantitative gap. The highest-lift rule discovered by NiaARM-PSO (lift = 63) involves antecedent intervals that fall between the quartile boundaries Apriori is forced to discretize on, which is precisely the regime that classical ARM cannot reach.

The interpretable form of the rule reads: when a video is younger than seven days, has a CTR between 25% and 35%, and an engagement rate between 25% and 35%, it falls in the viral views range with confidence 1.0 in this corpus. Such a rule offers an actionable, human-readable definition of flash virality on the Boburu channel, precisely the kind of pattern that is systematically missed by classical ARM, and our experiments confirm this argument empirically on a new domain.

PSO outperformed DE on rule count and average lift, which is consistent with broader observations that PSO's velocity-momentum mechanism explores narrower interval boundaries faster on small datasets.

3.5 Flash Viral Signature

The traffic-decomposition analysis exposes the temporal anatomy of the spike. Between February 22, 2026 (the day before the second viral video was published) and February 23, 2026 (the snapshot in which it dominates), cumulative views grew explosively—the canonical flash-viral profile. By the very next snapshot on February 24 the daily delta had collapsed to a handful of additional views, and by February 25 the channel had returned to baseline.

On the spike day itself, the algorithmic share of viral views was minimal: Where Boburu Phone produced only 1.0 algorithmic view (0.1%) against 1,296.0 external views (99.9%), and Boburu Didn't See Phone produced 3.0 algorithmic views (0.2%) against 1,597.0 external views (99.8%). The mean external share across the two viral videos is therefore 99.9%.

This represents an inverted profile compared to the standard trajectories documented in contemporary social media studies, in which algorithmic traffic typically dominates. The Boburu spike was therefore driven by an off-platform amplifier, most plausibly a re-share on a high-traffic external aggregator—and the YouTube recommender played a passive, lagging role rather than serving as the cause.

A second observation concerns thematic concentration. The two videos that generated 92.6% of channel views are both “Phone”-themed, and the highest-lift NiaARM rule contains the antecedent CTR $\in$ [25%, 35%], a range achieved by only those two videos.

The combination of thematic concentration, extreme external-traffic ratio, and sub-week age window jointly constitutes the flash-viral signature that the framework was designed to surface. While none of these three signals alone is diagnostic in isolation, but the conjunction—flagged simultaneously by the autoencoder and a NiaARM rule—yields zero false positives in our corpus.

3.6 Discussion and Implications

Three theoretical implications follow. First, autoencoders generalize to micro-channel virality detection even with as few as 40 training samples, provided that dropout regularization, early stopping, and a tight bottleneck are applied; this contradicts the common assumption that deep models require thousands of samples to outperform statistical baselines and aligns with the ADBench finding [14] that semi-supervised detectors are the most data-efficient. Second, nature-inspired numerical ARM is not merely a faster Apriori but a categorically different tool: it discovers rules whose antecedent intervals do not align with any pre-specified bin boundary, and these are the rules that carry the highest lift. Third, the decomposition of views into algorithmic and external components is a parsimonious yet under-used diagnostic that resolves an otherwise opaque virality event into a causal narrative.

Several practical implications follow for creators and platform analysts. The framework can run on a single CPU within minutes for channels of fewer than 1,000 videos, making it suitable for real-time creator-side analytics. The high-lift NiaARM rule directly suggests creative levers—sustaining CTR above 25% during the first week of upload and ensuring early external promotion—that are operationally actionable. For platforms, the same framework can flag channels whose external-traffic ratios deviate from population norms, complementing the network-centric anomaly detectors of Shajari and Agarwal [16] and the related commenter-behavior characterization by Shajari et al. [31] and Kirdemir et al. [32].

Limitations must be acknowledged. The corpus is restricted to a single channel and 42 active videos, and the two-video positive class is too small for cross-validated confidence intervals. The Autoencoder's performance ceiling is therefore likely under-estimated. NiaARM rule counts grow linearly with the iteration budget, and pruning by support × ift remains a manual step. Finally, the framework characterizes what a flash-viral video looks like ex post but does not yet predict virality ex ante, since the spike-day features are only available after the spike.

4. CONCLUSION

This study proposed and empirically validated a hybrid framework that pairs a semi-supervised Deep Learning Autoencoder with NiaARM, a nature-inspired numerical association-rule miner, to detect and characterize flash-viral videos on YouTube Shorts. The framework was instantiated on six temporal snapshots of the Indonesian Boburu channel [4], comprising 63 unique videos and 42 active videos, and benchmarked against six classical anomaly detectors and an Apriori ARM baseline. Three empirical findings stand out. First, the Autoencoder achieved an F1 of 0.667 with 100% recall, tying the best classical detector while outperforming Isolation Forest, LOF, One-Class SVM, and DBSCAN, a result that would have been impossible without semi-supervised training on 40 normal samples and a four-unit bottleneck. Second, NiaARM-PSO discovered 3,115 high-lift numerical association rules versus 43 produced by Apriori, with a maximum lift of 63.00 versus 2.52, an improvement of approximately 25 times; the dominant rule states that a video uploaded within seven days, with CTR and engagement rate both between 25% and 35%, falls in the viral views range with confidence 1.0, a pattern structurally invisible to Apriori. Third, the traffic-source decomposition revealed that 99.9% of viral views originated from external platforms rather than YouTube's own recommendation graph, attributing the spike to off-platform amplification rather than the recommender. The principal theoretical contribution is methodological: chaining a reconstruction-based detector with a numerical-ARM explainer produces both the what (which videos are anomalous) and the why (which feature intervals jointly characterize the anomaly), closing a long-standing gap between anomaly detection and rule mining on numerical data. The principal practical contribution is a creator-side analytics tool that runs on commodity hardware and yields actionable rules. Limitations include the single-channel scope, the small positive class, and the descriptive rather than predictive nature of the framework. Future work will extend evaluation to multiple channels and platforms (TikTok, Instagram Reels), introduce a temporal autoencoder (LSTM-AE) for ex-ante virality forecasting, integrate the NiaAutoARM AutoML pipeline of Mlakar et al. [29], augment the framework with explainable attention, and study adversarial robustness against engagement-fraud attacks.

REFERENCES

[1] C. Violot, T. Elmas, I. Bilogrevic, and M. Humbert, “Shorts vs. Regular Videos on YouTube: A Comparative Analysis of User Engagement and Content Creation Trends,” in *Proceedings of the 16th ACM Web Science Conference*, 2024, pp. 213–223. doi: 10.1145/3614419.3644023.

[2] J. Q. Wang, “Investigating the Effectiveness of Short Form Media Advertisements Compared to Long Form Media Advertisements,” Advances in Economics, Management and Political Sciences, vol. 89, no. 1, pp. 98–103, Jun. 2024, doi: 10.54254/2754-1169/89/20231451.

[3] R. Ramzan and S. Hashmat, “Short-Form Video Platforms and Their Impact on News Consumption & Civic Engagement: a Review,” International Journal of Social Sciences Bulletin, vol. 4, no. 2, pp. 474–491, Feb. 2026, doi: 10.5281/zenodo.18630606.

[4] Boburu, "Boburu Official," YouTube. [Online]. Available: https://youtube.com/@boburuofficial. [Accessed: April 20, 2026].

[5] A. S. Yaro, F. Maly, and P. Prazak, “Outlier Detection in Time-Series Receive Signal Strength Observation Using Z-Score Method with Sn Scale Estimator for Indoor Localization,” Applied Sciences, vol. 13, no. 6, p. 3900, Mar. 2023, doi: 10.3390/app13063900.

[6] M. Hubert and E. Vandervieren, “An adjusted boxplot for skewed distributions,” Comput Stat Data Anal, vol. 52, no. 12, pp. 5186–5201, Aug. 2008, doi: 10.1016/j.csda.2007.11.008.

[7] D. Alcan, K. Ozdemir, B. Ozkan, A. Y. Mucan, and T. Ozcan, “A Comparative Analysis of Apriori and FP-Growth Algorithms for Market Basket Analysis Using Multi-level Association Rule Mining,” in Industrial Engineering in the Covid-19 Era, F. Calisir and M. Durucu, Eds., Cham: Springer Nature Switzerland, 2023, pp. 128–137.

[8] M. Kaushik, R. Sharma, S. A. Peious, M. Shahin, S. ben Yahia, and D. Draheim, “A Systematic Assessment of Numerical Association Rule Mining Methods,” SN Comput Sci, vol. 2, no. 5, p. 348, 2021, doi: 10.1007/s42979-021-00725-2.

[9] G. Pang, C. Shen, L. Cao, and A. van den Hengel, “Deep Learning for Anomaly Detection,” ACM Comput Surv, vol. 54, no. 2, pp. 1–38, Mar. 2022, doi: 10.1145/3439950.

[10] S. Givnan, C. Chalmers, P. Fergus, S. Ortega-Martorell, and T. Whalley, “Anomaly Detection Using Autoencoder Reconstruction upon Industrial Motors,” Sensors, vol. 22, no. 9, p. 3166, Apr. 2022, doi: 10.3390/s22093166.

[11] B. X. Yong and A. Brintrup, “Do Autoencoders Need a Bottleneck for Anomaly Detection?,” IEEE Access, vol. 10, pp. 78455–78471, 2022, doi: 10.1109/ACCESS.2022.3192134.
[12] Ž. Stupan and I. F. Jr., “NiaARM: A minimalistic framework for Numerical Association Rule Mining,” J Open Source Softw, vol. 7, no. 77, p. 4448, Sep. 2022, doi: 10.21105/joss.04448.
[13] I. Fister, D. Fister, I. Fister, V. Podgorelec, and S. Salcedo-Sanz, “Time series numerical association rule mining variants in smart agriculture,” J Ambient Intell Humaniz Comput, vol. 14, no. 12, pp. 16853–16866, 2023, doi: 10.1007/s12652-023-04694-7.
[14] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, “ADBench: anomaly detection benchmark,” in Proceedings of the 36th International Conference on Neural Information Processing Systems, in NIPS ’22. Red Hook, NY, USA: Curran Associates Inc., 2022.
[15] S. Zannettou et al., “Analyzing User Engagement with TikTok’s Short Format Video Recommendations using Data Donations,” in Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, in CHI ’24. New York, NY, USA: Association for Computing Machinery, 2024. doi: 10.1145/3613904.3642433.
[16] S. Shajari and N. Agarwal, “Developing a network-centric approach for anomalous behavior detection on youtube,” Soc Netw Anal Min, vol. 15, no. 3, pp. 1–16, Mar. 2025, doi: 10.1007/s13278-025-01417-y.
[17] S. Darrab et al., “Anomaly Detection Algorithms: Comparative Analysis and Explainability Perspectives,” in Data Science and Machine Learning, D. Benavides-Prado, S. Erfani, P. Fournier-Viger, Y. L. Boo, and Y. S. Koh, Eds., Singapore: Springer Nature Singapore, 2024, pp. 90–104.
[18] Y. Kim and M. Vasarhelyi, “Anomaly detection with the density based spatial clustering of applications with noise (DBSCAN) to detect potentially fraudulent wire transfers,” The International Journal of Digital Accounting Research, vol. 24, pp. 57–91, 2024, doi: 10.4192/1577-8517-v24_3.
[19] J. A. Diaz-Garcia, M. D. Ruiz, and M. J. Martin-Bautista, “A survey on the use of association rules mining techniques in textual social media,” Artif Intell Rev, vol. 56, pp. 1175–1200, 2023, doi: 10.1007/s10462-022-10196-3.
[20] H. Xu, Q. Deng, Z. Zhang, and S. Lin, “A hybrid differential evolution particle swarm optimization algorithm based on dynamic strategies,” Sci Rep, vol. 15, 2025, doi: 10.1038/s41598-024-82648-5.
[21] G. Vrbančič, L. Brezočnik, U. Mlakar, D. Fister, and I. Fister Jr., “NiaPy: Python microframework for building nature-inspired algorithms,” J Open Source Softw, vol. 3, no. 23, p. 613, Mar. 2018, doi: 10.21105/joss.00613.
[22] M. Owusu-Adjei, J. ben Hayfron-Acquah, T. Frimpong, and G. Abdul-Salaam, “Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems,” PLOS Digital Health, vol. 2, no. 11, p. e0000290, Nov. 2023, doi: 10.1371/journal.pdig.0000290.
[23] H. Xu, G. Pang, Y. Wang, and Y. Wang, “Deep Isolation Forest for Anomaly Detection,” IEEE Trans Knowl Data Eng, vol. 35, no. 12, pp. 12591–12604, 2023, doi: 10.1109/TKDE.2023.3270293.
[24] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, and M. Salehi, “Deep Learning for Time Series Anomaly Detection: A Survey,” ACM Comput Surv, vol. 57, no. 1, pp. 1–42, Jan. 2025, doi: 10.1145/3691338.
[25] S. Maleki, S. Maleki, and N. R. Jennings, “Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering,” Appl Soft Comput, vol. 108, p. 107443, 2021, doi: https://doi.org/10.1016/j.asoc.2021.107443.
[26] A. Telikani, A. H. Gandomi, and A. Shahbahrami, “A survey of evolutionary computation for association rule mining,” Inf Sci (N Y), vol. 524, pp. 318–352, 2020, doi: https://doi.org/10.1016/j.ins.2020.02.073.
[27] G. Li, T. Wang, Q. Chen, P. Shao, N. Xiong, and A. Vasilakos, “A Survey on Particle Swarm Optimization for Association Rule Mining,” Electronics (Basel), vol. 11, no. 19, p. 3044, Sep. 2022, doi: 10.3390/electronics11193044.
[28] U. Mlakar, I. Fister, and I. Fister, “Variable-Length Differential Evolution for Numerical and Discrete Association Rule Mining,” IEEE Access, vol. 12, pp. 4239–4254, 2024, doi: 10.1109/ACCESS.2023.3348408.
[29] U. Mlakar, I. Fister, and I. Fister, “NiaAutoARM: Automated Framework for Constructing and Evaluating Association Rule Mining Pipelines,” Mathematics, vol. 13, no. 12, p. 1957, Jun. 2025, doi: 10.3390/math13121957.
[30] M. Kaushik, R. Sharma, I. Fister Jr., and D. Draheim, "Numerical Association Rule Mining: A Systematic Literature Review," arXiv preprint, arXiv:2307.00662, Jul. 2023, doi: 10.48550/arXiv.2307.00662.
[31] S. Shajari, M. Alassad, and N. Agarwal, "Commenter Behavior Characterization on YouTube Channels," arXiv preprint, arXiv:2304.07681, Apr. 2023, doi: 10.48550/arXiv.2304.07681.
[32] B. Kirdemir, O. Adeliyi, and N. Agarwal, "Towards Characterizing Coordinated Inauthentic Behaviors on YouTube," in Proc. 2nd Workshop on Reducing Online Misinformation through Credible Information Retrieval (ROMCIR 2022), co-located with ECIR 2022, CEUR Workshop Proceedings, vol. 3138, pp. 100–116, Apr. 2022.

