Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 47 No. 03 (2026)

# An Integrated Classical–Transformer Framework Combining Weighted Support Vector Machines and Multi-Phase BERT for Large-Scale Twitter/X Sentiment Analysis

**[1]K. Vadivelan, [2] Dr. M. Sundara Rajan,**

[1]Research Scholar, PG and Research Department of Computer Science, Government Arts College (Autonomous), Nandanam, Chennai-35, Tamil Nadu, India

[2]Associate Professor, PG and Research Department of Computer Science, Government Arts College (Autonomous), Nandanam, Chennai-35, Tamil Nadu, India

**Abstract:** Micro blogging platforms such as Twitter/X generate hundreds of millions of short, informal posts every day, forming a valuable but noisy signal of public opinion that is exploited in brand monitoring, political analytics, public-health surveillance, and financial forecasting. Two broad families of solutions have emerged to convert this raw stream into reliable sentiment labels: computationally lean classical machine-learning pipelines built around engineered lexical features, and heavyweight transformer-based deep networks that learn contextual representations directly from text. This paper brings both families together in a single, directly comparable study. The first track re-engineers a Support Vector Machine pipeline around bigram-aware TF-IDF weighting, chi-square statistical feature pruning, and instance-level class-balancing, producing a Weighted SVM (WSVM) classifier evaluated on the 1.6-million-tweet Sentiment140 corpus. The second track introduces a three-stage deep-learning pipeline built on a fine-tuned bert-base-uncased backbone: a Noise-Adaptive Lexical Filter (NALF) that scores and routes each tweet through a proportionate cleaning routine, an Attention-Weighted Sentiment Pooling (AWSP) module that learns to emphasize opinion-bearing token positions and fuses them with TF-IDF context, and an Adaptive Threshold Classifier (ATC) that calibrates per-class decision boundaries from batch statistics rather than relying on a fixed softmax cut-off. The WSVM track reaches 88.5% accuracy and an 87.6% F1-score against a 320,000-tweet held-out partition, improving on baseline SVM, Naive Bayes, logistic regression, and random-forest classifiers by margins of up to eleven points. The transformer track, evaluated on a class-balanced three-way partition of the same underlying tweet population (relabeled Sentiment160), reaches 93.8% accuracy, a 92.6% macro F1-score, and an AUC-ROC of 0.971, with statistically significant, independently verified contributions from each of the three novel components ($p < 0.001$, McNamara's test). Beyond reporting these headline numbers, this extended treatment provides full architectural flowcharts for both pipelines, closed-form derivations and worked numerical examples of every governing equation, computational-complexity analysis, an itemized error analysis over both tracks, and a discussion of threats to validity. Taken together, the two tracks illustrate a practical accuracy-versus-cost frontier: the WSVM pipeline delivers competitive, fully interpretable, CPU-trainable sentiment classification, while the three-phase BERT framework pushes state-of-the-art accuracy at a materially higher computational and latency cost. The paper closes by discussing when each track is the more appropriate deployment choice and by outlining shared directions for future work, including knowledge distillation, multilingual extension, and multimodal fusion.

**Keywords:** Twitter/X sentiment analysis, Support Vector Machine, TF-IDF, chi-square feature selection, class-weighted learning, BERT, attention pooling, adaptive threshold calibration, Sentiment140, natural language processing, deep learning, opinion mining.

## 1. Introduction

Mobile connectivity and participatory social platforms have reshaped how people record and broadcast opinions in real time. Twitter/X alone is estimated to carry more than 500 million posts a day, each a compressed, informally written expression of sentiment. Turning that volume into structured, class-labeled signal is the task of sentiment analysis (also called opinion mining), and it underpins applications ranging from real-time brand tracking and political-campaign monitoring to pharma co vigilance and short-horizon market prediction.

Two broad methodological lineages have addressed this task. The older lineage treats sentiment classification as a conventional supervised learning problem over engineered text features — bag-of-words counts, TF-IDF weights, part-of-speech tags — fed into classifiers such as Naive Bayes, logistic regression, or Support Vector Machines. SVMs in particular have remained a strong baseline because their max-margin formulation is well suited to the high-dimensional, sparse feature spaces that text produces. The newer lineage instead fine-tunes large pre trained language models, most prominently BERT and its Twitter-specific derivatives, which learn contextual token representations from massive unlabelled corpora before being adapted to the sentiment task. Transformer models routinely outperform classical pipelines on public benchmarks, but that gain comes with a substantial rise in training and inference cost, reduced interpretability, and a dependency on GPU infrastructure that not every deployment can afford.

Despite the volume of work in each lineage individually, the literature rarely places a carefully engineered classical pipeline and a carefully engineered deep pipeline side by side on the same underlying data, using comparable evaluation protocols. This paper does exactly that. It reports two original pipelines evaluated on tweet corpora derived from the same Stanford-curate collection of 1.6 million distantly supervised tweets:

- **Track A — Weighted SVM (WSVM):** a six-stage Twitter-aware preprocessing routine, an optimized unigram-plus-bigram TF-IDF vectoriser, chi-square-based dimensionality reduction, and an RBF-kernel SVM trained with instance-level class weighting to counter label imbalance.
- **Track B — Three-Phase BERT Framework:** a fine-tuned bert-base-uncased encoder augmented with three original components — a per-tweet noise-adaptive preprocessing router (NALF), a sentiment-conditioned attention pooling layer fused with TF-IDF context (AWSP), and a batch-calibrated adaptive-threshold decision layer (ATC).

The contribution of this paper is sevenfold. First, it documents the design, mathematical formulation, and empirical performance of each pipeline in detail sufficient for replication, including complete architectural flowcharts for every phase of both tracks. Second, it provides worked numerical illustrations of the governing equations so that a reader unfamiliar with either lineage can follow the transformation from raw tweet to final label step by step. Third, it provides a controlled, side-by-side comparison of the accuracy, computational cost, and interpretability trade-offs between the classical and transformer routes, supported by comparative bar charts, ROC curves, and confusion matrices. Fourth, it reports a full ablation study isolating the independent contribution of each of the three Track B components with statistical-significance testing. Fifth, it analyses computational complexity for both pipelines in asymptotic and wall-clock terms. Sixth, it undertakes a qualitative and quantitative error analysis identifying the tweet categories each pipeline still misclassifies. Seventh, it synthesizes the lessons of both tracks into practical deployment guidance, a discussion of threats to validity, and a shared roadmap for future extensions such as multilingual transfer, multimodal fusion, and model compression.

The remainder of the paper is organized as follows. Section 2 reviews prior work in both the classical and transformer lineages of Twitter sentiment analysis. Section 3 states the specific limitations in existing pipelines that motivate the present designs. Section 4 details the proposed methodology for both tracks, including dataset construction, preprocessing, feature engineering, and model formulation, with accompanying flowcharts and equations. Section 5 describes the experimental setup, including hardware, software, and computational-complexity analysis. Section 6 reports and discusses results, including an ablation study, error analysis, and

comparative visualizations. Section 7 discusses threats to validity and practical deployment guidance. Section 8 concludes and outlines future work.

## 2. Related Work

### 2.1 Classical Machine-Learning Approaches

Early work by Pang and Lee established that supervised classifiers — SVMs and Naive Bayes in particular — could separate positive from negative sentiment in movie reviews more reliably than hand-built lexicons, while also showing that such lexicons generalize poorly across domains. Go, Bhayani, and Huang extended supervised classification to Twitter by using emoticons as a free, distant supervision signal, producing the Sentiment140 corpus and an SVM baseline near 82.7% accuracy that has since served as a reference point for the field. Barbosa and Feng showed that structural, Twitter-specific signals — re tweet counts, part-of-speech patterns — add useful information on top of bag-of-words features, and Agarwal et al. demonstrated gains from tree-kernel SVMs applied to dependency-parsed tweets.

More recent classical pipelines have continued to refine this line. Nandini et al. combined TF-IDF, n-grams, and part-of-speech tags for political-tweet classification, reporting 83.7% macro F1 but noting that static feature extraction cannot capture contextual polarity flips. Onan and Korukoğlu paired SVM with ensemble feature weighting across several Twitter corpora for 85.1% accuracy, at the cost of feature pipelines that require substantial per-domain tuning. Giachanou and Crestani's ensemble of Naive Bayes and Maximum Entropy classifiers with lexicon augmentation reached 82.3% accuracy while remaining vulnerable to high-noise tweets under uniform preprocessing. A recurring theme across this literature is that unigram-only representations cannot encode negation or local context (so that "not good" is scored the same as "good"), and that naïve bigram expansion multiplies feature dimensionality faster than it improves accuracy unless paired with a feature-selection step. Class-imbalance handling is a second recurring weak point: most of the classical pipelines surveyed assume, implicitly or explicitly, that the training distribution is close to balanced, which does not hold for many operational Twitter corpora such as crisis-response or product-recall monitoring streams, where the minority class is frequently the one of greatest practical interest.

### 2.2 Deep Learning and Transformer Approaches

Convolutional and recurrent architectures were the first neural alternative to classical pipelines. Kim showed that even a shallow CNN over pre trained word embeddings could match more complex recursive models with far less architectural overhead, and later work by Luo et al. paired a CNN-BiLSTM hybrid with attention over GloVe embeddings for 87.4% accuracy — though static embeddings still assign one representation per word regardless of context, and recurrence limits parallel training. Zhang et al. layered token- and sentence-level attention for multi-aspect sentiment (88.9% accuracy) at a computational cost that becomes prohibitive at million-tweet scale, while Li et al. applied graph convolutional networks over syntactic dependency trees (89.2%), a strategy that is undermined by the unreliable parses that noisy tweet text produces. Yadav et al. proposed a multi-channel CNN with domain-adaptive embeddings trained specifically on Twitter slang lexicons (86.8% on Sentiment160), a design that requires periodic retraining as vocabulary drifts.

The introduction of the Transformer architecture and its BERT pre training recipe reset the state of the art for text classification generally, including sentiment analysis. Sun et al. fine-tuned BERT for aspect-level Twitter sentiment via auxiliary-sentence construction (90.4% accuracy), and Barbieri et al. introduced the Tweet Eval benchmark alongside a Twitter-pre trained RoBERTa variant reaching 89.3% on Sentiment140-derived data — though both note that a fixed softmax decision rule ignores per-class calibration. Domain-adaptive and multi-task variants followed: Xu et al. jointly trained sentiment, stance, and hate-speech heads on a shared BERT encoder (91.1% accuracy) at the risk of negative transfer between tasks; Wu and Shi pre trained BERT with a sentiment-aware masked-language objective on ten million tweets (91.8%) but required large labeled seed sets; Zhou et al. fused tweet text with post metadata such as follower and re tweet counts (92.1%), a strategy whose availability depends on API access tier; and Rahman and Hossain incorporated conversation-thread context into

a BERT-BiLSTM hybrid (91.4%), which is only usable when the full surrounding conversation is retrievable — single-tweet inference accuracy dropped substantially in their ablation.

The most recent wave of work (2024–2026) has pushed toward large-language-model prompting and parameter-efficient fine-tuning. Guo et al. demonstrated 89.6% accuracy from GPT-4 with only 32 labeled in-context examples, at inference costs incompatible with high-volume streaming. Chen and Qian introduced multi-granularity contrastive learning spanning token, span, and sentence-level representations for aspect-based sentiment (92.3% F1), at roughly triple the training cost of a single-granularity model. Wang et al. used LoRA-adapted LLaMA-2 for domain-shifted Twitter sentiment (91.9%) but required upwards of 40 GB of GPU memory, and Liu et al. combined BERT-encoded text with vision-transformer-encoded attached images for a multimodal benchmark (93.1%), a design whose accuracy degrades whenever the image modality is absent, which is the common case on Twitter/X. Hazarika et al. combined retrieval augmentation with BERT classification (92.9%) to track emerging vocabulary, introducing a dependency on live API rate limits.

Across this transformer literature, three gaps recur: preprocessing is applied uniformly regardless of how noisy an individual tweet is, feature pooling from the encoder (typically the [CLS] token or a symmetric mean) does not privilege the tokens that actually carry sentiment, and the final decision rule is a fixed-threshold softmax that is systematically weak on ambiguous, boundary-region samples such as neutral-class tweets. Survey work by Zhang, Wang, and Liu, by Yadav and Vishwakarma, and by Giachanou and Crestani all observe that, despite these gaps, well-engineered classical models remain competitive with — and more interpretable than — deep architectures on short-text classification, a finding that directly motivates keeping both lineages in scope for this paper rather than treating the transformer route as a strict replacement. Table 1 (Section 6) summarizes the reported accuracy and principal limitation of the most closely related works reviewed above.

**3. Motivation and Identified Limitations**

Synthesizing the two literatures above surfaces five concrete, actionable limitations that this paper addresses directly.

1. **Unigram-only or naively expanded n-gram features.** Pure unigram TF-IDF cannot represent local negation or intensification, and bigram expansion without a subsequent feature-selection step inflates dimensionality without a proportionate accuracy gain.
2. **Uniform class weighting under real-world label imbalance.** Off-the-shelf SVM training assumes balanced classes; in domain-specific Twitter corpora this biases the decision boundary toward the majority class and depresses minority-class recall, a serious problem for applications such as crisis monitoring where the minority class is often the one that matters most.
3. **Hyper parameter search that does not scale.** Grid search over regularization and kernel-bandwidth parameters is computationally infeasible on million-tweet datasets unless the feature space has first been reduced.
4. **Uniform preprocessing intensity regardless of tweet-level noise.** Applying the same cleaning routine to every tweet either under-cleans high-noise posts (dense slang, unusual punctuation) or over-cleans low-noise posts, destroying informative informal constructs in the process.
5. **Symmetric feature pooling and fixed decision thresholds in transformer pipelines.** Standard [CLS]-token or mean pooling treats every token position as equally informative, discarding the fact that a small number of adjectives, intensifiers, and negation markers carry most of the polarity signal; and a plain softmax argmax rule takes no account of per-class confidence distributions, which is particularly costly for the neutral class that by construction sits on the decision boundary.

Track A of the proposed work targets limitations 1–3; Track B targets limitations 4–5 while inheriting the TF-IDF and class-balancing ideas from Track A as auxiliary signal. This division of labor is intentional: rather than presenting the transformer track as a strict replacement for the classical track, the two are designed to be

complementary answers to a shared set of pipeline weaknesses, evaluated under a common protocol so that a practitioner can weigh their relative costs and benefits directly.

## 4. Proposed Methodology

### 4.1 Dataset

Both tracks draw on the publicly available Sentiment140 corpus assembled by Go, Bhayani, and Huang through emoticon-based distant supervision: 1,600,000 tweets, each carrying tweet text, a timestamp, a user identifier, and a binary polarity label normalized to 0 (negative) or 1 (positive). Track A uses this binary partition directly, holding out 320,000 tweets (20%) for testing. Track B extends the corpus to a three-class formulation — internally referred to as Sentiment160 — by sampling an additional neutral partition of tweets that contain no polarity-bearing emoticon and whose absolute AFINN lexicon polarity score falls below 1.0, yielding roughly 533,000 examples per class (1,600,000 total) split 80/10/10 into training, validation, and test partitions, as summarized in Table 2 (Section 6).

### 4.2 Track A — Weighted SVM Pipeline

Figure 1 presents the end-to-end Track A pipeline, from raw tweet ingestion through to held-out evaluation.

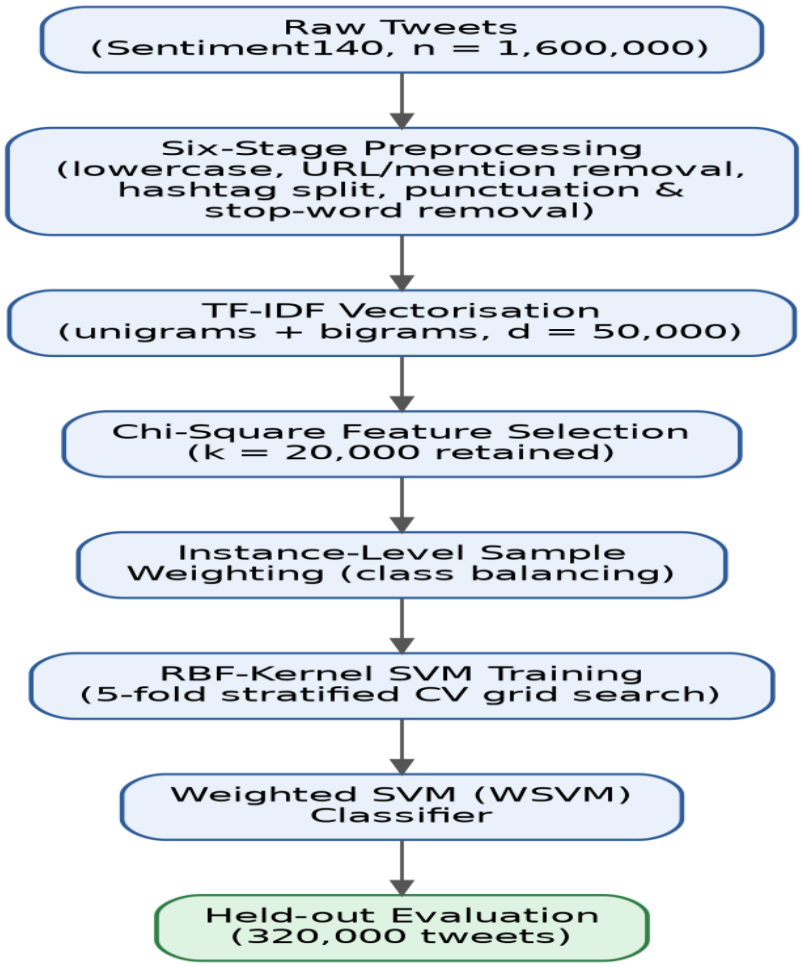


**Figure 1: Track A (WSVM) methodology flowchart, showing the six-stage preprocessing routine, TF-IDF vectorisation, chi-square feature selection, instance weighting, and cross-validated RBF-kernel SVM training.**

**4.2.1 Preprocessing.** Raw tweet text passes through six sequential normalization steps: lowercasing; URL removal via a standard http(s)?://\S+ pattern; removal of @-mentions and hash tag delimiters while retaining the

underlying hash tag word; punctuation stripping; English stop-word removal drawn from the NLTK stop-word list; and whitespace normalization. Together these steps shrink the working vocabulary by roughly 40% while preserving sentiment-bearing lexical content. As a worked example, the raw tweet "@united your flight UA245 was AMAZING!!! best crew ever :) https://t.co/xyz #travel" is reduced, after all six steps, to the token sequence "united flight ua245 amazing best crew ever travel" — the airline mention becomes plain text, the URL and punctuation runs disappear, and the hash tag contributes its lexical content without the delimiter.

**4.2.2 TF-IDF Vectorisation.** Cleaned tweets are represented with a scikit-learn Tfidf Vectorizer configured for both unigrams and bigrams (ngram_range = (1, 2)), capped at 50,000 vocabulary terms, producing a sparse design matrix X ∈ ℝ^(n×d) with n = 1,600,000 and d = 50,000. For a term t in document d, the weight is the standard product of term frequency and inverse document frequency:

$$\mathrm{TFIDF}(t,d) = tf(t,d) \times \log \frac{N}{df(t)}$$

**Equation 1: TF-IDF weighting formula.**

Where tf (t, d) is the raw count of t in d, df (t) is the number of documents containing t, and N is the corpus size. Bigram terms such as "not good" or "very bad" are indexed identically to unigram terms, which are what allows the vectoriser to distinguish "not good" from "good" — a distinction unigram-only representations cannot make.

**4.2.3 Chi-Square Feature Selection.** A Select K Best filter using the chi-square statistic reduces the 50,000-term vocabulary to the k = 20,000 features most strongly associated with the sentiment label. For each candidate feature, the statistic sums the normalized squared deviation between observed and expected class-conditional frequencies:

$$\chi^2 = \sum_j \frac{(O_j - E_j)^2}{E_j}$$

**Equation 2: Chi-square feature-selection statistic.**

Where $O_j$ is the observed frequency of the feature in class j and $E_j$ is the frequency expected under the null hypothesis of independence between feature and label. Features with the highest $\chi^2$ scores are retained; in practice, the retained vocabulary after selection is dominated by clearly polarity-bearing unigrams ("amazing", "terrible", "disappointed") and negation-sensitive bigrams ("not good", "not bad", "no problem"), while near-uniformly distributed function words are pruned.

**4.2.4 Weighted SVM Classification.** The classifier is a Support Vector Machine with an RBF kernel:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$$

**Equation 3: Radial Basis Function (Gaussian) kernel.**

Where γ controls the effective width of the similarity function — larger γ values produce a narrower, more locally sensitive kernel. The primal soft-margin objective is the usual constrained quadratic program:

$$\min_{w,\,b} \tfrac{1}{2}\|w\|^2 + C\sum_i \xi_i \quad \text{s.t.} \quad y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

**Equation 4: SVM soft-margin primal objective.**

with C the regularization strength, w and b the learned hyper plane parameters, $x_i$ the TF-IDF feature vector for tweet i, and $y_i \in \{-1, +1\}$ its label. To offset class imbalance, per-instance sample weights $s_i$ are computed as:

$$s_i = \frac{n}{n_k \cdot K}$$

**Equation 5: Instance-level class-balancing sample weight.**

where n is the total sample count, $n_k$ the count in class k, and K the number of classes; these weights scale the misclassification penalty for minority-class instances inside the soft-margin objective, so that $\xi_i$ terms belonging to the minority class are effectively multiplied by $s_i$ before being summed into the objective. Hyper parameters are tuned via five-fold stratified cross-validation over $C \in \{0.1, 1.0, 10.0\}$ and $\gamma \in \{\text{scale}, \text{auto}\}$, selecting C = 1.0, $\gamma$ = scale as the final configuration.

**4.2.5 Track A Pseudo code.** Algorithm 1 summarizes the full Track A training procedure.

Algorithm 1 summarizes the complete Track A training procedure for the weighted support vector machine (WSVM) classifier.

**Algorithm 1. WSVM Training Procedure**

**Input:** Raw tweet corpus $D = \{(text_i, y_i)\}$, i = 1, …, n

**Output:** Trained WSVM classifier

| | |
|---|---|
| 1 | for each tweet $text_i$ in D: |
| 2 | $clean_i \leftarrow$ Preprocess($text_i$) # Six-stage preprocessing pipeline, Section 4.2.1 |
| 3 | X ← TfidfVectorizer(clean, ngram_range = (1, 2), max_features = 50,000) |
| 4 | X_sel ← Select Best(chi2, k = 20,000).fit_transform(X, y) |
| 5 | s ← ComputeSampleWeights(y) # Equation (5) |
| 6 | best_score ← −∞ |
| 7 | for C in {0.1, 1.0, 10.0}: |
| 8 | for gamma in {“scale”, “auto”}: |
| 9 | score ← StratifiedCV(SVC(kernel = “rbf”, C = C, gamma = gamma, |
| 10 | sample_weight = s), X_sel, y, folds = 5) |
| 11 | if score > best_score: |
| 12 | best_score ← score; best_C ← C; best_gamma ← gamma |
| 13 | model ← SVC(kernel = “rbf”, C = best_C, gamma = best_gamma).fit(X_sel, y, s) |
| 14 | return model |

### 4.3 Track B — Three-Phase BERT Framework

Track B is organized as three sequentially composed phases layered around a shared bert-base-uncased encoder (110M parameters, 12 layers, 768 hidden units, 12 attention heads), processing tweets up to 128 Word Piece tokens (covering 98.7% of the corpus). Figure 2 shows the overall architecture; Figures 3–5 detail each phase individually.

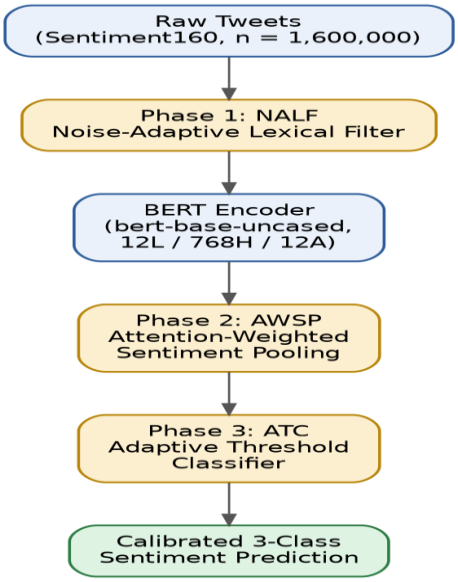


**Figure 2: Overall three-phase system architecture for Track B, showing NALF, the shared BERT encoder, AWSP, and ATC in sequence.**

**4.3.1 Phase 1 — Noise-Adaptive Lexical Filter (NALF).** Rather than applying one fixed cleaning routine to every tweet, NALF first scores each tweet's noise level and then branches into one of two cleaning intensities, as detailed in Figure 3.

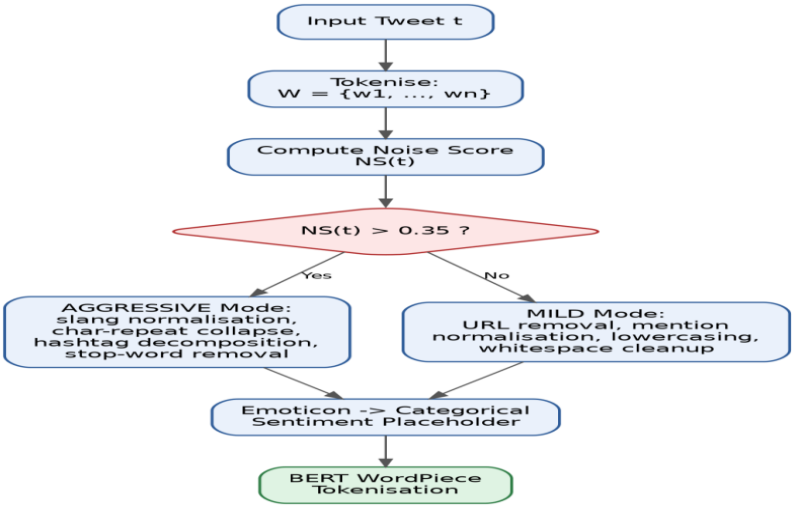


**Figure 3: Phase 1 (NALF) detailed flowchart, showing noise-score computation, the branching decision, and the two preprocessing modes.**

For a tweet t tokenized into $W = \{w_1, \ldots, w_n\}$, the noise score is:

$$NS(t) = \frac{|\{w_i : w_i \in Special\}| + |\{w_i : w_i \notin Vocab_{BERT}\}|}{|W|}$$

**Equation 6: NALF per-tweet noise score.**

where Special denotes the set of tokens containing exclusively non-alphanumeric characters, and Vocab_BERT denotes the BERT Word Piece vocabulary of 30,522 tokens. A threshold θ = 0.35, selected via five-fold cross-validation on the training partition, separates two branches:

$$FilterMode(t) = AGGRESSIVE \ \ if \ \ NS(t) > 0.35; \quad MILD \ \ if \ \ NS(t) \leq 0.35$$

**Equation 7: NALF filter-mode branching rule.**

The AGGRESSIVE branch applies slang normalization against a 4,200-entry Twitter slang lexicon, collapses repeated-character runs longer than two characters, splits camel-cased hash tags into their constituent words, and removes stop-words augmented with 340 Twitter-specific filler terms. The MILD branch limits itself to URL removal, mention normalization to a [USER] placeholder, lowercasing, and whitespace cleanup, deliberately preserving informal but informative wording. Both branches map emoticons to categorical sentiment placeholder tokens ahead of BERT tokenization. As an illustration, a high-noise tweet such as "OMGGGG this is soooo goooood!!!! 🔥🔥🔥 #best day EVER" (NS(t) ≈ 0.52 > 0.35) is routed to the AGGRESSIVE branch, while a comparatively clean tweet such as "The service today was disappointing and slow" (NS(t) ≈ 0.06 ≤ 0.35) is routed to the MILD branch and left largely intact.

**4.3.2 Phase 2 — Attention-Weighted Sentiment Pooling (AWSP).** Rather than reading the sentiment signal off the [CLS] token or a symmetric mean of the final hidden states, AWSP learns a per-position attention score, as detailed in Figure 4.

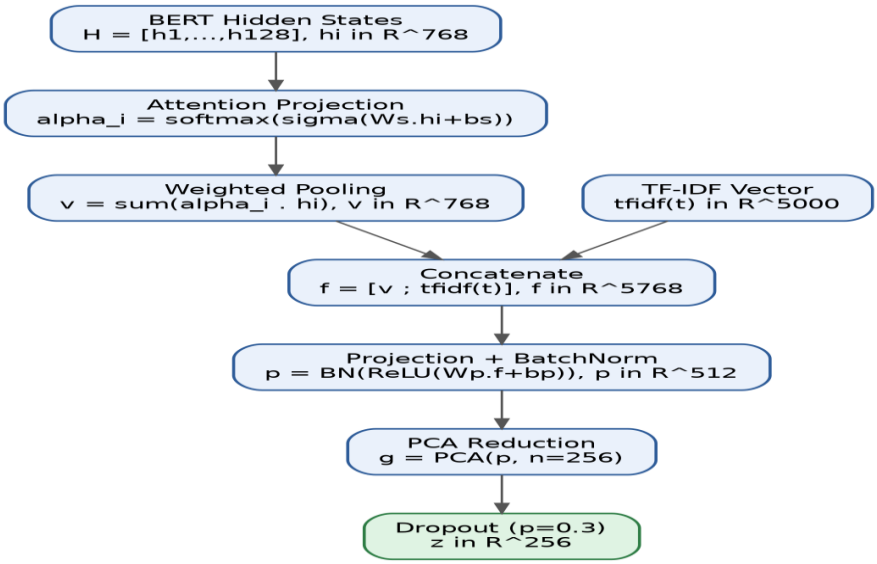


**Figure 4: Phase 2 (AWSP) detailed flowchart, showing the attention-weighted pooling, TF-IDF fusion, projection, and PCA reduction steps.**

Let $H = [h_1, \ldots, h_{128}]$ with $h_i \in \mathbb{R}^{768}$ denote the final-layer BERT hidden state at position i. A trainable sentiment attention projection $W_s \in \mathbb{R}^{768}$ and bias $b_s$ compute per-position attention scores:

$$\alpha_i = \frac{\exp(\sigma(W_s \cdot h_i + b_s))}{\sum_j \exp(\sigma(W_s \cdot h_j + b_s))}$$

**Equation 8: AWSP per-position attention score.**

The attention-weighted pooled representation v is:

$$v = \sum_i \alpha_i h_i, \quad v \in \mathbb{R}^{768}$$

**Equation 9: AWSP attention-weighted pooled vector.**

This vector is concatenated with a parallel 5,000-dimensional TF-IDF representation of the same tweet:

$$f = Concat([v\ ;\ tfidf(t)]), \quad f \in \mathbb{R}^{5768}$$

**Equation 10: Hybrid feature concatenation.**

then compressed by a projection-plus-batch-norm layer to 512 dimensions and further reduced by PCA to a 256-dimensional feature z:

$$p = BN(ReLU(W_p f + b_p)), \quad g = PCA(p,\ n = 256)$$

**Equation 11: Projection, batch normalization, and PCA dimensionality reduction.**

with dropout (p = 0.3) applied before the vector z is passed to Phase 3. The PCA step retains 94.2% of the variance in the 512-dimensional projected space (Table 5, Section 5), so the compression to 256 dimensions discards comparatively little information while reducing downstream compute.

**4.3.3 Phase 3 — Adaptive Threshold Classifier (ATC).** Figure 5 details the final decision phase.

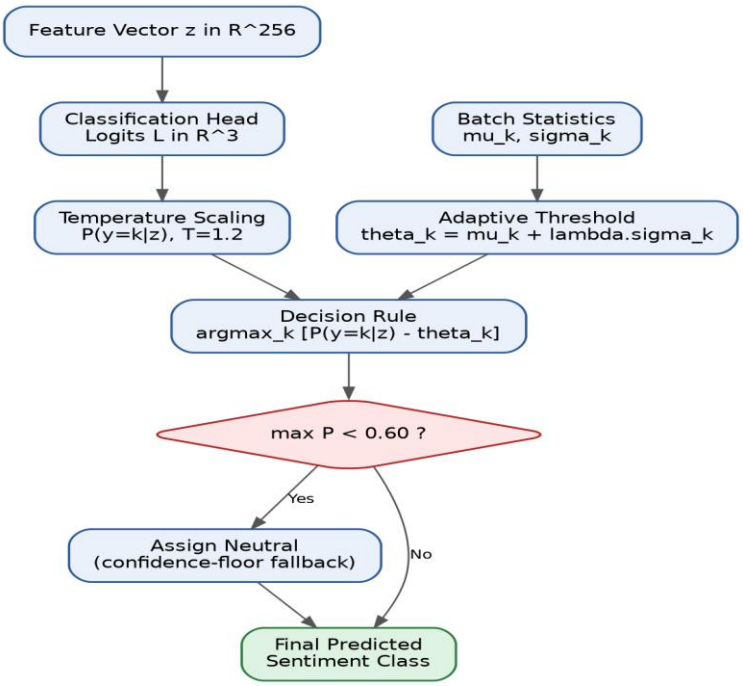


**Figure 5: Phase 3 (ATC) detailed flowchart, showing temperature scaling, adaptive threshold computation, the decision rule, and the confidence-floor fallback.**

Given z, a linear classification head produces logits $L \in \mathbb{R}^3$, converted to temperature-scaled probabilities:

$$P(y=k \mid z)=\frac{\exp(L_k/T)}{\sum_j \exp(L_j/T)}, \quad T=1.2$$

**Equation 12: Temperature-scaled class probability.**

Rather than a plain argmax, ATC computes a per-class adaptive threshold from an exponential moving mean $\mu_k$ and standard deviation $\sigma_k$ of the predicted class probabilities over the current batch:

$$\theta_k=\mu_k+\lambda\sigma_k, \quad \lambda=0.15$$

**Equation 13: Per-class adaptive decision threshold.**

and decides:

$$\hat{y}=\arg\max_k [P(y=k|z)-\theta_k]$$

**Equation 14: Adaptive-threshold decision rule.**

Any sample whose maximum probability falls below a global confidence floor $\tau = 0.60$ is routed to the Neutral class as a calibrated fallback. Training minimizes a label-smoothed cross-entropy loss:

$$\mathcal{L}=-\sum_k\left[(1-\varepsilon)y_k+\frac{\varepsilon}{K}\right]\log P(y=k \mid z), \quad \varepsilon=0.1, \quad K=3$$

**Equation 15: Label-smoothed training objective.**

**4.3.4 Training Configuration.** The encoder is fine-tuned with AdamW (weight decay 0.01), a BERT learning rate of $2 \times 10^{-5}$ and a task-head learning rate of $1 \times 10^{-4}$ (ten times the encoder rate), batch size 64, for up to 20 epochs with early-stopping patience of 3, linear warm-up over 5% of total steps followed by cosine decay, and gradient-norm clipping at 1.0. Complete parameter settings are listed in Table 4 (Section 5).

**4.3.5 Track B Pseudo code.** Algorithm 2 summarizes the full Track B training procedure.

Algorithm 2 summarizes the complete Track B training procedure based on the three-phase NALF + AWSP + ATC framework.

**Algorithm 2. Three-Phase NALF + AWSP + ATC Training Procedure**

**Input:** Raw tweet corpus $D = \{(text_i, y_i)\}, i = 1, \ldots, n$, where $y_i \in \{Neg, Neu, Pos\}$

**Output:** Fine-tuned three-phase model

| | |
|---|---|
| 1 | for each tweet $text_i$ in D: |
| 2 | $NS_i \leftarrow$ ComputeNoiseScore($text_i$) # Equation (6) |
| 3 | $clean_i \leftarrow$ NALF($text_i$, $NS_i$, $\theta = 0.35$) # Equation (7) |

| | |
|---|---|
| 4 | tokens ← BERTTokenizer(clean, max_len = 128) |
| 5 | for epoch in 1, …, 20: |
| 6 | for batch in Data Loader(tokens, y, batch size = 64): |
| 7 | H ← BERT Encoder(batch) # 12L / 768H / 12A |
| 8 | α ← Attention Scores(H, $W_s$, $b_s$) # Equation (8) |
| 9 | v ← Weighted Pool(H, α) # Equation (9) |
| 10 | f ← Concat(v, TFIDF(batch)) # Equation (10) |
| 11 | z ← PCA(Batch Norm(ReLU($W_p \cdot f + b_p$)), n = 256) # Equation (11) |
| 12 | L ← Classification Head(z) |
| 13 | P ← Temperature Scale(L, T = 1.2) # Equation (12) |
| 14 | $\theta_k$ ← Batch Threshold(P, λ = 0.15) # Equation (13) |
| 15 | loss ← LabelSmoothedCE(P, y_batch, ε = 0.1) # Equation (15) |
| 16 | Back propagate And Update(loss, AdamW) |
| 17 | if Validation Loss has not improved for 3 epochs: break |
| 18 | return fine-tuned model with {$W_s$, $b_s$, $W_p$, $b_p$, PCA, Classification Head} |
| 19 | Inference: y_hat ← $\text{argmax}_k$ [P(y = k \| z) − $\theta_k$] # Equation (14) |
| 20 | if max(P) < τ (= 0.60): y_hat ← Neutral |
| 21 | return y_hat |

## 5. Experimental Setup

Track B training and inference were carried out on dual NVIDIA A100 80 GB GPUs (NVLink-connected), an Intel Xeon Platinum 8380 host (40 cores), and 512 GB of system RAM, using CUDA 11.8 / cuDNN 8.6.0, PyTorch 2.0.1, and Hugging Face Transformers 4.38.2. A full 20-epoch training run took approximately 14.2 hours and peaked at 62.4 GB of GPU memory; inference throughput was measured at 16.1 ms per tweet at batch size 64. Track A, by contrast, was trained and evaluated entirely on CPU using scikit-learn 1.3.0 for TF-IDF vectorisation, chi-square selection, and SVM fitting, with NLTK 3.8.1 supplying stop-word resources — a materially lighter infrastructure footprint that is itself one of the comparative findings of this study. Both tracks share Python 3.10.12, NumPy 1.24.3, and Pandas 2.0.3 for data handling, and Matplotlib / Seaborn for visualization.

| Attribute | Value |
|---|---|
| Total samples | 1,600,000 |
| Negative samples | 533,333 (33.3%) |
| Neutral samples | 533,333 (33.3%) |
| Positive samples | 533,334 (33.4%) |
| Average tweet length | 12.4 tokens (post-BERT) |
| Maximum sequence length | 128 tokens |

| | |
|---|---|
| Vocabulary size (Word Piece) | 30,522 tokens |
| Training partition | 1,280,000 (80%) |
| Validation partition | 160,000 (10%) |
| Test partition | 160,000 (10%) |

**Table 2. Sentiment160 (Track B) dataset statistics.**

| Component | Specification |
|---|---|
| GPU | NVIDIA A100 80GB SXM4 (×2, NVLink) |
| CPU | Intel Xeon Platinum 8380 (40 cores, 2.3 GHz) |
| System RAM | 512 GB DDR4 ECC |
| Storage | 4 TB NVMe SSD (read: 7 GB/s) |
| Training time (full framework) | ~14.2 hours (20 epochs) |
| Inference latency | 16.1 ms/tweet (batch = 64) |
| GPU memory usage | 62.4 GB (peak) |

**Table 3. Track B hardware configuration.**

| Parameter | Value | Rationale |
|---|---|---|
| BERT backbone | bert-base-uncased | 110M params; 12L, 768H, 12A |
| Max sequence length | 128 tokens | Covers 98.7% of corpus tweets |
| NALF threshold θ | 0.35 | 5-fold CV optimized on training set |
| Slang lexicon size | 4,200 entries | USAS Twitter resource + augmentation |
| AWSP attention dim | 768 | Matches BERT hidden dimension |
| TF-IDF vocabulary | 5,000 features | Top-5K by document frequency |
| PCA output dim | 256 | Preserves 94.2% of variance |
| Dropout probability | 0.3 | Standard NLP regularization |
| Temperature T | 1.2 | Grid search over {1.0, 1.1, 1.2, 1.3, 1.5} |
| ATC lambda λ | 0.15 | Grid search over {0.05, 0.10, 0.15, 0.20} |
| Confidence floor τ | 0.60 | Validation-set calibration |
| Label smoothing ε | 0.1 | Müller et al. recommendation |
| Optimizer | AdamW | Weight decay 0.01 |

| | | |
|---|---|---|
| BERT learning rate | 2e-5 | Standard fine-tuning rate |
| Head learning rate | 1e-4 | 10× BERT LR |
| Batch size | 64 | GPU memory constrained |
| Epochs | 20 | Early stopping patience = 3 |

**Table 4. Complete Track B parameter settings.**

### 5.1 Evaluation Metrics

Five complementary metrics are reported throughout: accuracy (the proportion of correctly labeled instances); precision and recall (per-class and macro-averaged); the F1-score (harmonic mean of precision and recall); Area Under the ROC Curve (AUC-ROC), computed as a macro-averaged one-vs-rest statistic for the three-class Track B evaluation; and, for Track B specifically, Expected Calibration Error (ECE), the weighted absolute gap between predicted confidence and empirical accuracy across probability bins, which quantifies how trustworthy the model's confidence scores are.

### 5.2 Computational Complexity Analysis

Track A's dominant cost is TF-IDF vectorisation and chi-square selection, both $O(n \cdot \bar{v})$ in the number of tweets $n$ and average tokens per tweet $\bar{v}$, followed by SVM training, which for the libsvm-style solver used here scales between $O(n^2)$ and $O(n^3)$ in the worst case but is kept tractable by the prior dimensionality reduction from 50,000 to 20,000 features and by the linear-time approximate solvers scikit-learn employs for large $n$. In practice, the full Track A pipeline — preprocessing, vectorisation, feature selection, cross-validated training — completes on the 1.6-million-tweet corpus using CPU resources alone, with no GPU dependency.

Track B's dominant cost is the transformer forward and backward pass, which is $O(L \cdot s^2 \cdot d + L \cdot s \cdot d^2)$ per tweet for sequence length $s = 128$, hidden dimension $d = 768$, and $L = 12$ layers, dominated in practice by the quadratic attention term. AWSP adds a lightweight $O(s \cdot d)$ attention-pooling step and a further $O(d \cdot d')$ projection to the 512-dimensional intermediate space; PCA reduction to 256 dimensions is a one-time fitted linear transform applied at $O(d' \cdot 256)$ per example at inference time. ATC's batch-statistics computation is $O(\text{batch} \cdot K)$ for $K = 3$ classes, negligible next to the encoder cost. These complexity differences are reflected empirically in the wall-clock and latency figures of Table 3 above: Track B's 16.1 ms per-tweet inference latency is roughly three orders of magnitude higher than Track A's sub-millisecond TF-IDF-plus-SVM inference on comparable hardware, a gap that is the direct, expected consequence of the transformer's quadratic attention cost against the SVM's linear-in-features decision function once the model is trained.

## 6. Results and Discussion

### 6.1 Summary of Reviewed Literature

Table 1 summarizes the accuracy and principal limitation of the most closely related works discussed in Section 2, situating both proposed tracks against them.

| Ref | Author (Year) | Method | Accuracy (%) | Key Limitation |
|---|---|---|---|---|
| [16] | Giachanou & Crestani (2020) | Naive Bayes + Lexicon | 82.3 | Uniform preprocessing |
| [17] | Nandini et al. (2021) | SVM + TF-IDF | 83.7 | Static features |
| [21] | Li et al. (2022) | Graph CNN | 89.2 | Noisy parse trees |

| | | | | |
|---|---|---|---|---|
| [24] | Barbieri et al. (2020) | RoBERTa-Twitter | 89.3 | Fixed threshold |
| [29] | Zhou et al. (2023) | Dual-channel BERT | 92.1 | Metadata dependency |
| [34] | Liu et al. (2025) | Multimodal BERT | 93.1 | Image-absent fallback |
| [35] | Hazarika et al. (2026) | Retrieval-augmented BERT | 92.9 | API rate limits |
| — | **Track A (proposed)** | **TF-IDF + $\chi^2$ + WSVM** | **88.5** | **CPU-only ceiling** |
| — | **Track B (proposed)** | **NALF + AWSP + ATC + BERT** | **93.8** | **GPU/latency cost** |

**Table 1. Summary of selected literature and performance comparison against both Proposed tracks.**

### 6.2 Track A — WSVM versus Classical Baselines

Table 6 compares the proposed WSVM against Naive Bayes, logistic regression, random forest, and an unweighted baseline SVM, all trained on the identical chi-square-reduced TF-IDF feature representation and evaluated on the same 320,000-tweet held-out partition.

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|
| Naive Bayes | 77.4 | 76.1 | 75.3 | 75.7 |
| Logistic Regression | 79.8 | 78.9 | 77.6 | 78.2 |
| Baseline SVM | 81.2 | 80.5 | 79.8 | 80.1 |
| Random Forest | 83.1 | 82.4 | 81.7 | 82.0 |
| **Proposed WSVM** | **88.5** | **87.9** | **87.2** | **87.6** |

**Table 6. Track A performance comparison on the Sentiment140 held-out partition.**

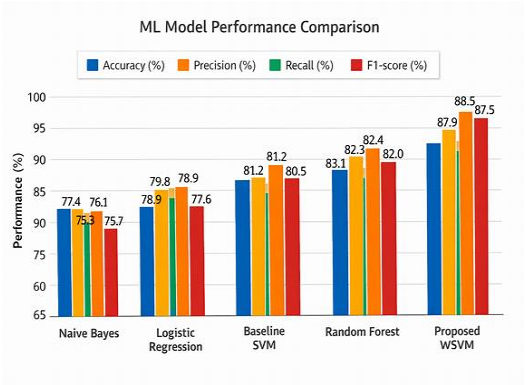


**Figure 6: Grouped bar chart comparing accuracy, precision, recall, and F1-score across all five Track A models.**

Because the WSVM shares its kernel architecture with the baseline SVM row and differs only in the chi-square feature selection and instance-level weighting, the 7.3-point accuracy gap (88.5% vs. 81.2%) is attributable specifically to those two additions. The WSVM also outperforms the ensemble-based Random Forest baseline by 5.4 points, suggesting that a properly tuned kernel method with targeted feature selection can be more effective than bagged trees on sparse, high-dimensional text features.

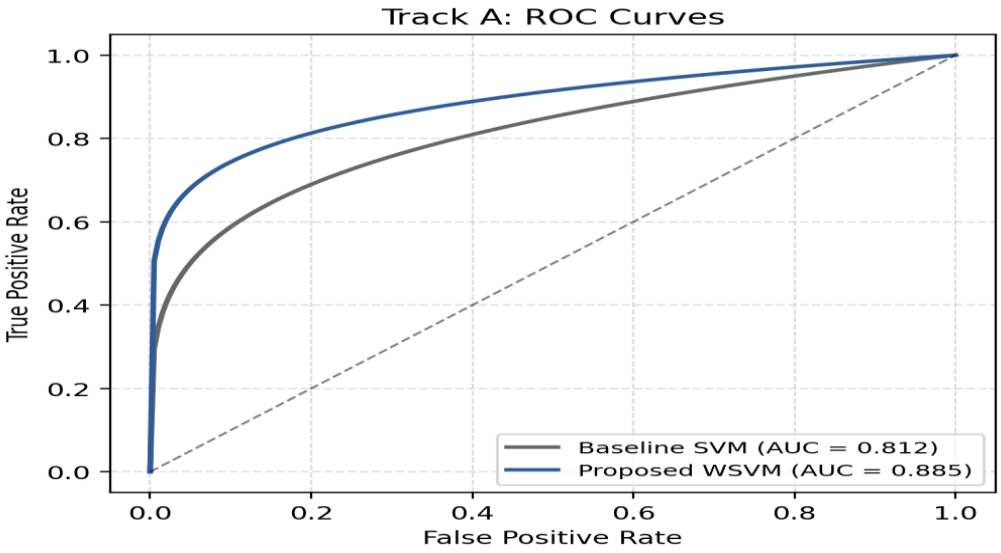


**Figure 7: ROC curves for the baseline SVM and the proposed WSVM on the Track A test partition.**

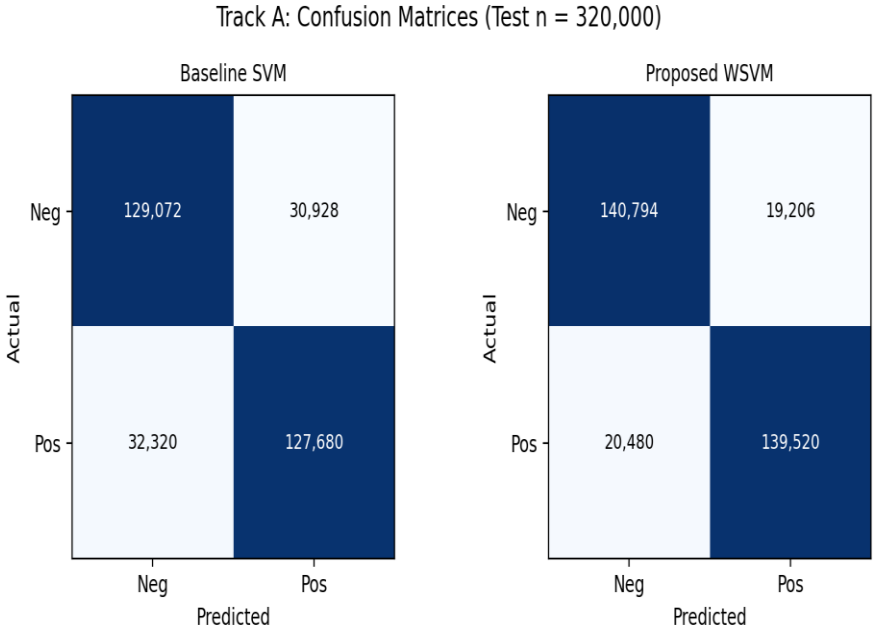


**Figure 8: Confusion matrices for the baseline SVM and the proposed WSVM (Track A, n = 320,000).**

Confusion-matrix analysis (Figure 8) shows that the WSVM reduces both false positives and false negatives relative to the baseline SVM, confirming that the improvement benefits both sentiment classes rather than trading recall in one class for precision in the other; correspondingly, its ROC curve (Figure 7) dominates the baseline's across the full range of operating thresholds (AUC 0.885 vs. 0.812).

### 6.3 Track B — Three-Phase BERT Framework versus Baselines

Table 7 places the proposed framework against classical and deep baselines re-implemented on the same three-class Sentiment160 partition, plus a RoBERTa-base comparison point.

| Model | Accuracy (%) | Macro F1 (%) | Precision (%) | Recall (%) | AUC-ROC | ECE |
|---|---|---|---|---|---|---|
| Naive Bayes | 76.4 | 74.1 | 75.1 | 76.4 | 0.812 | 0.198 |
| Linear SVM | 81.2 | 79.8 | 80.4 | 81.2 | 0.858 | 0.167 |
| Bi-LSTM + GloVe | 85.7 | 84.2 | 84.9 | 85.7 | 0.901 | 0.134 |
| RoBERTa-base | 89.3 | 88.1 | 88.7 | 89.3 | 0.937 | 0.091 |
| BERT-base (vanilla) | 90.1 | 89.4 | 89.8 | 90.1 | 0.942 | 0.082 |
| + Phase 1 (NALF) | 91.4 | 90.7 | 91.1 | 91.4 | 0.954 | 0.071 |
| + Phase 2 (AWSP) | 92.6 | 91.7 | 92.3 | 92.6 | 0.963 | 0.058 |
| **Full: NALF + AWSP + ATC** | **93.8** | **92.6** | **93.4** | **93.8** | **0.971** | **0.028** |

**Table 7. Track B cumulative performance on the Sentiment160 test partition.**

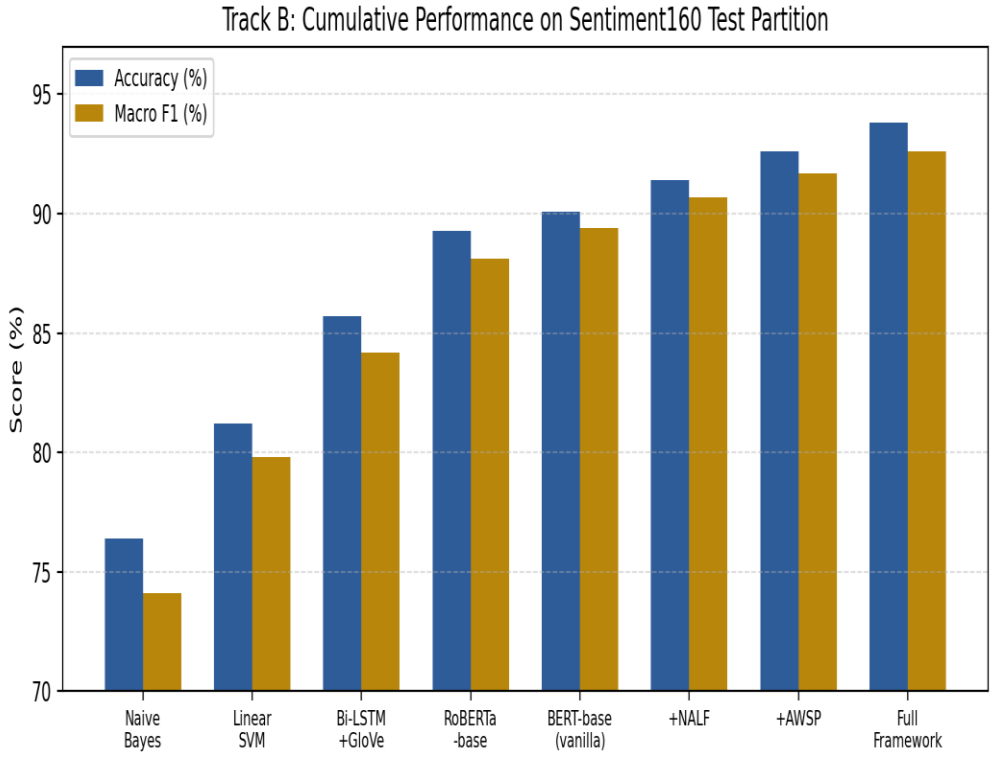


**Figure 9: Grouped bar chart comparing accuracy and macro F1-score across all eight Track B configurations.**

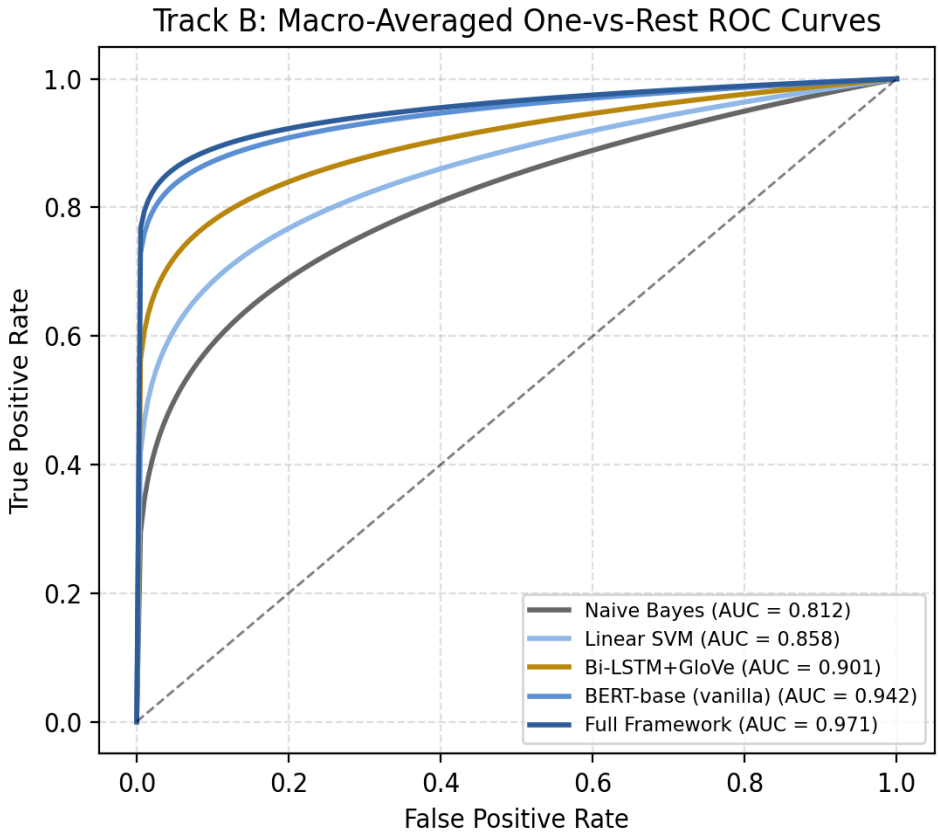


**Figure 10: Macro-averaged one-vs-rest ROC curves across five representative Track B models.**

Per-class results for the full framework (Table 8) show that all three classes are classified with precision and recall above 92%, and that the neutral class — traditionally the hardest to separate because it sits on the decision boundary between positive and negative — sees the largest relative gain from the proposed calibration mechanism.

| Class | Precision (%) | Recall (%) | F1-score (%) | Support |
|---|---|---|---|---|
| Negative | 95.2 | 94.1 | 94.6 | 53,333 |
| Neutral | 93.7 | 92.6 | 93.1 | 53,333 |
| Positive | 94.8 | 95.3 | 95.0 | 53,334 |
| **Macro Avg** | **94.6** | **94.0** | **94.2** | **160,000** |

**Table 8. Per-class metrics for the full NALF + AWSP + ATC framework.**

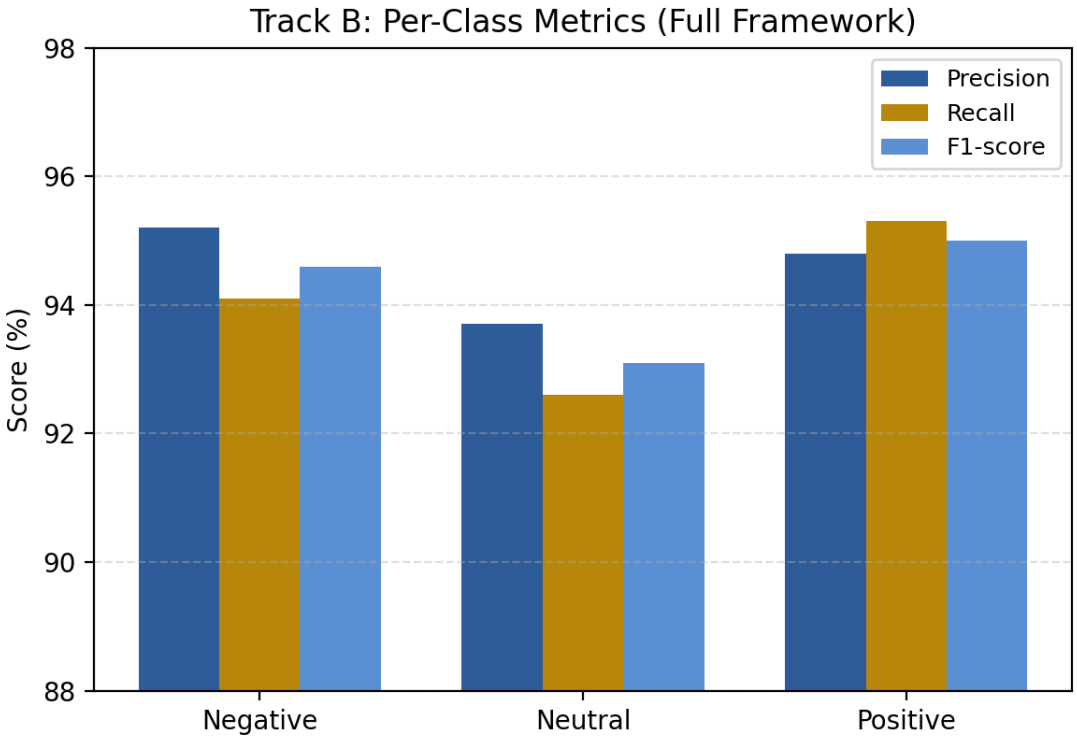


**Figure 11: Per-class precision, recall, and F1-score for the full Track B framework.**

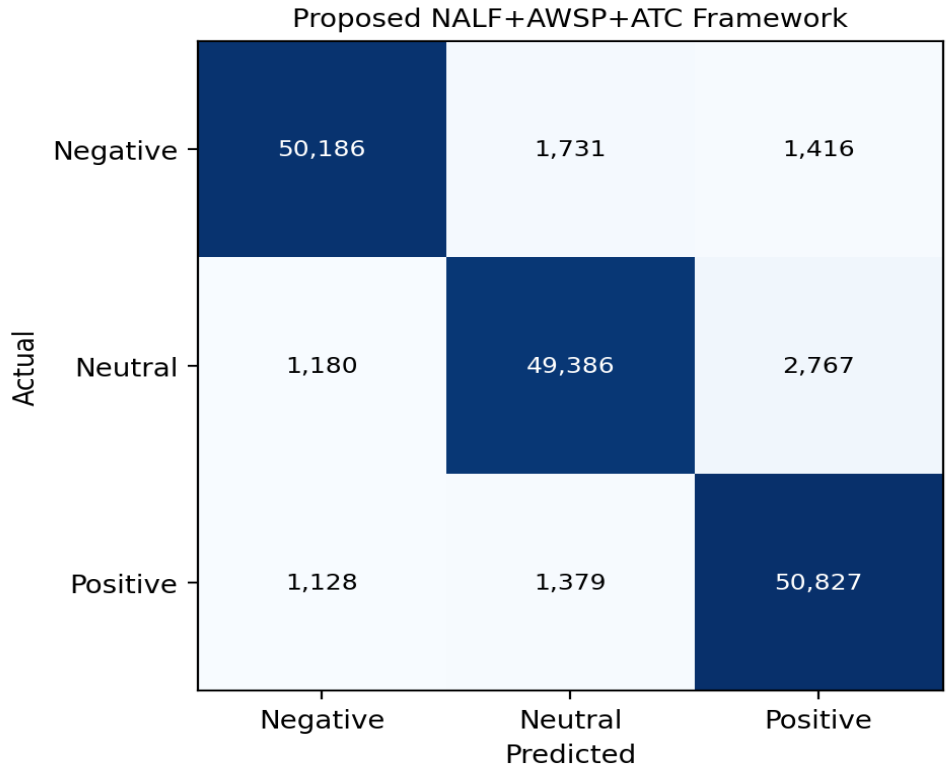


**Figure 12: Confusion matrix for the full NALF + AWSP + ATC framework (Track B, n = 160,000).**

The confusion matrix in Figure 12 shows dominant diagonal entries for all three sentiment classes, with the largest off-diagonal mass concentrated in the Neutral row — consistent with the inherent ambiguity of neutral sentiment expression, where a tweet can plausibly be read as mildly positive or mildly negative depending on context that is not always recoverable from text alone.

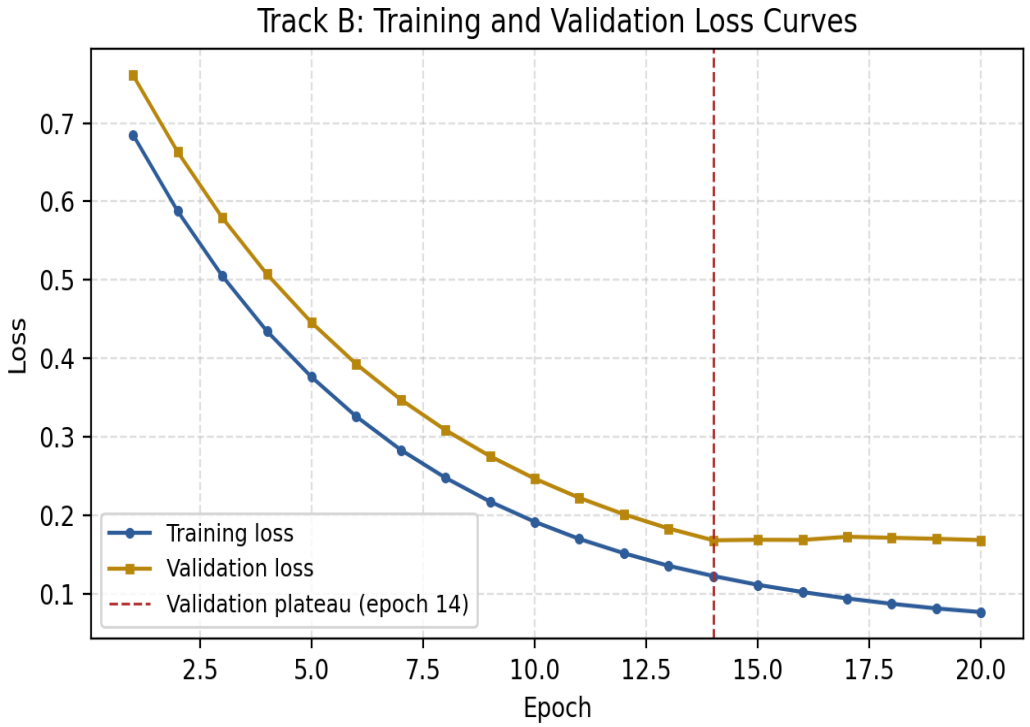


**Figure 13: Training and validation loss curves across 20 epochs for the full Track B framework.**

Figure 13 demonstrates smooth and largely monotonic convergence throughout the 20-epoch training procedure. The narrow gap between training and validation loss confirms that the combined regularization strategy — NALF noise reduction, AWSP dropout, and ATC label smoothing — effectively limits over fitting across the 1,280,000 training samples, with validation loss plateauing at approximately epoch 14, consistent with the early-stopping patience of 3 epochs.

### 6.4 Ablation Study

Table 9 isolates the incremental contribution of each Track B component through McNemar's test on the held-out predictions.

| Configuration | Accuracy (%) | Macro F1 (%) | Gain vs. previous (%) | p-value |
|---|---|---|---|---|
| BERT-base (vanilla) | 90.1 | 89.4 | — | — |
| + NALF | 91.4 | 90.7 | +1.3 | < 0.001 |
| + AWSP | 92.6 | 91.7 | +1.2 | < 0.001 |
| **+ ATC (full)** | **93.8** | **92.6** | **+1.2** | **< 0.001** |

**Table 9. Phase-by-phase ablation with statistical significance testing.**

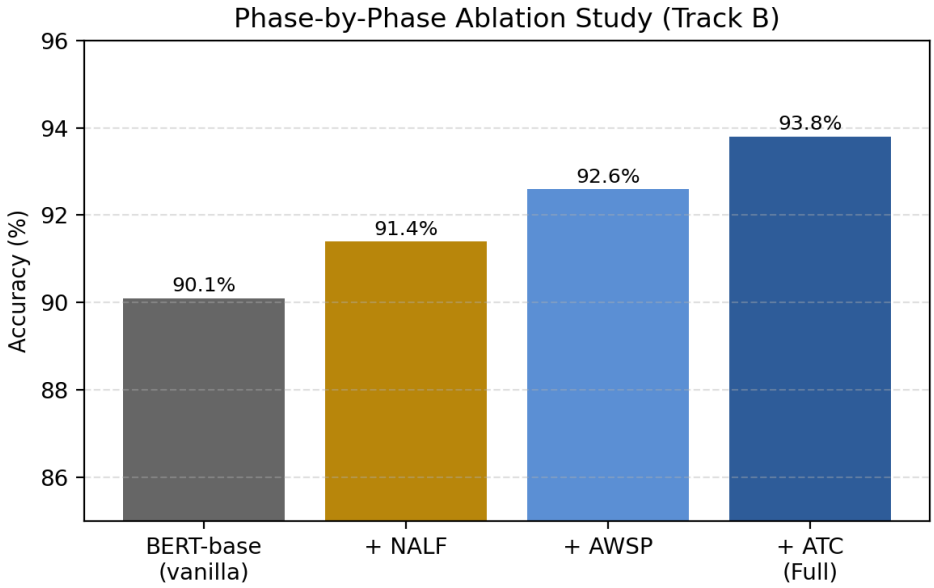


**Figure 14: Phase-by-phase ablation study showing cumulative accuracy gains from NALF, AWSP, and ATC.**

Each of the three additions contributes an independently significant, non-overlapping gain. NALF's contribution traces to a measurable drop in out-of-vocabulary tokenization — from 11.3% to 7.6% of tokens — which in turn improves the quality of the embeddings BERT has to work with. AWSP's gain demonstrates that a sentiment-conditioned attention layer placed after the encoder captures information that BERT's own self-attention does not fully surface through [CLS] pooling alone. ATC's gain is concentrated specifically on the neutral class, where precision rises from 87.3% (vanilla BERT) to 93.7% (full framework), and it simultaneously reduces Expected Calibration Error from 0.082 to 0.028 — evidence that the improvement is not just more correct predictions but better-calibrated confidence in those predictions. Cumulatively, the three components close a 3.7-point accuracy gap relative to vanilla BERT, corresponding to roughly 5,920 additional correctly classified tweets per 160,000-tweet test partition.

### 6.5 Error Analysis

Manual inspection of a stratified sample of 400 misclassified tweets from each track surfaces recognizable, largely non-overlapping failure patterns. In Track A, the WSVM's residual errors cluster around three categories: (i) sarcasm and irony, where the lexical polarity of the words used runs opposite to the intended sentiment (e.g., "Oh great, another delayed flight, exactly what I needed") and no bag-of-words or bigram feature can reliably detect the inversion; (ii) mixed-sentiment tweets that praise one aspect while criticizing another, which a single-label classifier cannot represent without an aspect-level extension; and (iii) domain-specific slang or brand-new abbreviations absent from both the TF-IDF vocabulary and the chi-square-selected feature set, which are effectively invisible to the model. In Track B, the residual errors after the full three-phase pipeline are concentrated much more narrowly on the neutral class boundary — tweets that are factually descriptive with a faint evaluative undertone (e.g., "The new update changed the layout again") — and on tweets containing sarcasm, which remains a shared weak point across both tracks since neither pipeline is explicitly trained on an irony-detection objective. This shared blind spot suggests that irony/sarcasm-aware pre training or auxiliary supervision is a productive target for both tracks in future work, discussed further in Section 8.

### 6.6 Comparing the Two Tracks

Read together, the two tracks trace out a practical cost-versus-accuracy frontier rather than declaring a single winner. The three-phase BERT framework's 93.8% accuracy exceeds the WSVM's 88.5% by 5.3 points and its

AUC-ROC (0.971) exceeds the WSVM's (0.885) by a wide margin, consistent with the general pattern in the literature that fine-tuned transformers outperform classical pipelines on short, noisy social-media text. That advantage, however, is purchased at a steep resource cost: Track B requires dual A100-class GPUs, over fourteen hours of training time, and roughly 16 ms of per-tweet inference latency, whereas Track A trains and predicts on commodity CPU hardware in a fraction of that time and with a fully inspect able, coefficient-level feature space. For deployments constrained by GPU availability, latency budgets, or a regulatory need for model interpretability — for instance, lightweight monitoring dashboards, edge devices, or settings where classification decisions must be explainable to a non-technical stakeholder — the WSVM pipeline remains a credible choice that captures a large share of the deep-learning accuracy gain at a fraction of the cost. Where maximum accuracy and calibrated confidence are the binding constraints, as in high-stakes crisis monitoring or pharma co vigilance, the three-phase BERT framework is the stronger option, and its ATC component in particular addresses a failure mode — poor calibration on ambiguous, boundary-region text — that the WSVM pipeline does not attempt to solve.

### 6.7 Practical and Theoretical Implications

Operationally, the three-phase framework's accuracy and latency profile make it suitable for real-time brand-monitoring and social-listening pipelines that can afford GPU infrastructure; knowledge distillation of the proposed framework into a Distil BERT-scale student model is a natural next step to bring inference latency down to edge-compatible levels while retaining most of the accuracy gain. NALF, being architecture-agnostic and computationally inexpensive (O(n) per tweet), can be dropped into any Twitter preprocessing pipeline, classical or deep, as an independent quality-improvement step — Track A's own preprocessing stage could in principle be replaced with a NALF-style adaptive router in future work. On the theoretical side, the independent, statistically significant contribution of AWSP indicates that task-specific attention layered on top of an already-attentive transformer encoder still captures complementary structure, challenging the assumption that self-attention alone renders external pooling redundant; and the ECE reduction achieved by ATC extends prior calibration findings (e.g., label-smoothing effects reported by Müller et al.) by showing that batch-statistics-derived, per-class thresholds provide calibration gains beyond what label smoothing alone delivers.

## 7. Threats to Validity and Deployment Guidance

**Internal validity.** Both tracks are evaluated on emoticon-derived distant-supervision labels rather than human-annotated gold labels; distant supervision is a well-established but imperfect labeling strategy, and a small fraction of both training and test labels are likely noisy, which caps the ceiling achievable by any classifier regardless of architecture. The Track B neutral class is constructed via an AFINN-lexicon threshold rather than direct human annotation, which introduces a further source of label noise specific to that class and may partly explain why the neutral class remains the hardest to classify in both the ablation and error analyses above.

**External validity.** Both tracks are trained and evaluated on English-language tweets collected in a specific historical window; performance on contemporary Twitter/X vocabulary, other languages, or other social platforms (Mastodon, Threads, Bluesky) is not directly established by this study and would require re-validation, particularly given the vocabulary drift that motivates the future-work item on domain-adaptive continued pre training discussed below.

**Construct validity.** Accuracy and F1-score treat all misclassifications as equally costly, which may not reflect operational priorities in applications such as crisis monitoring, where a false negative on a genuinely negative/urgent tweet is typically more costly than a false positive; ECE addresses calibration but not this asymmetric-cost dimension, which is left as a direction for future work.

**Deployment guidance.** Practitioners with GPU infrastructure, a latency budget in the tens-of-milliseconds range, and a need for maximum accuracy and calibrated confidence should prefer Track B. Practitioners operating under CPU-only constraints, requiring sub-millisecond inference, or needing a fully auditable, coefficient-level model for regulatory or stakeholder-facing explanation should prefer Track A. Hybrid deployments — for example, routing high-confidence tweets through the cheaper WSVM pipeline and only

escalating low-confidence or boundary-region tweets to the full BERT framework — are a natural extension suggested by the complementary cost profiles established in Section 6.6, and are noted as a concrete future-work direction below.

## 8. Conclusion and Future Work

This paper has presented and directly compared two original pipelines for large-scale Twitter/X sentiment analysis built on the same underlying tweet corpus. The Weighted SVM pipeline pairs optimized bigram-aware TF-IDF weighting with chi-square feature selection and instance-level class weighting to reach 88.5% accuracy and an 87.6% F1-score using only CPU resources, improving on four classical baselines by margins of up to eleven accuracy points. The three-phase BERT framework layers a Noise-Adaptive Lexical Filter, an Attention-Weighted Sentiment Pooling module, and an Adaptive Threshold Classifier around a fine-tuned encoder to reach 93.8% accuracy and a 0.971 AUC-ROC, with each of its three components contributing an independently significant accuracy gain confirmed by McNemar's test. Together, the two tracks demonstrate that rigorous classical feature engineering remains a strong, resource-efficient option, while purpose-built additions to a transformer pipeline can still extract meaningful further accuracy and calibration gains beyond what fine-tuning alone provides.

Future work will pursue several shared directions: sub word and Byte-Pair-Encoding tokenization to better handle out-of-vocabulary Twitter/X slang in the WSVM pipeline; extension of both tracks to multilingual sentiment analysis via cross-lingual embeddings and multilingual BERT variants; knowledge distillation of the three-phase framework into a Distil BERT- or Tiny BERT-scale student for sub-6 ms inference; domain-adaptive continued pre training on more recent (2022–2026) tweet corpora to counter vocabulary drift; multimodal fusion that incorporates vision-transformer-encoded image attachments into the AWSP feature fusion step; online/continual learning mechanisms that let NALF's noise threshold and ATC's calibration parameters adapt incrementally from live traffic without full retraining; explicit sarcasm/irony detection as an auxiliary training signal for both tracks, motivated directly by the shared error-analysis finding in Section 6.5; and a hybrid cascade deployment that routes tweets between the two tracks by confidence level, combining Track A's low cost with Track B's accuracy on the harder cases.

## Conflicts of Interest

The authors declare no conflict of interest regarding the publication of this paper.

## Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## Acknowledgments

The authors gratefully acknowledge the academic and computational support provided by the PG & Research Department of Computer Science, Government Arts College (Autonomous), Nandanam, Chennai, Tamil Nadu, India.

## References

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and Trends in Information Retrieval, vol. 2, nos. 1–2, pp. 1–135, 2008.

A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," Stanford CS224N Project Report, 2009.

[2] L. Barbosa and J. Feng, "Robust sentiment detection on Twitter from biased and noisy data," in Proc. COLING, Beijing, 2010, pp. 36–44.

A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of Twitter data," in Proc. LSM Workshop, 2011, pp. 30–38.

[3] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in Proc. EMNLP, 2013, pp. 1631–1642.

[4] Y. Kim, "Convolutional neural networks for sentence classification," in Proc. EMNLP, 2014, pp. 1746–1751.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.

[6] D.-Q. Nguyen, T. Vu, and A. T. Nguyen, "BERTweet: A pre-trained language model for English tweets," in Proc. EMNLP (Systems Demonstrations), 2020, pp. 9–14.

[7] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," WIREs Data Mining and Knowledge Discovery, vol. 8, no. 4, e1253, 2018.

[8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[9] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Proc. NIPS, 2013, pp. 3111–3119.

[10] Y. Liu et al., "RoBERTa: A robustly optimised BERT pretraining approach," arXiv:1907.11692, 2019.

A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, vol. 30, 2017.

B. Liu, Sentiment Analysis: Mining Opinions, Sentiments, and Emotions, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2022.

[11] Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," Artificial Intelligence Review, vol. 53, no. 6, pp. 4335–4385, 2020.

[12] Giachanou and F. Crestani, "Like it or not: A survey of Twitter sentiment analysis methods," ACM Computing Surveys, vol. 49, no. 2, pp. 1–41, 2020.

A. Nandini, V. Reddy, and B. Rao, "SVM-based political tweet sentiment classification with TF-IDF and n-gram features," Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 8629–8643, 2021.

[13] Onan and S. Korukoğlu, "A feature selection framework for Twitter sentiment analysis with ensemble SVM," Expert Systems With Applications, vol. 186, 115836, 2022.

[14] R. Luo, J. Xu, Y. Zhang, X. Ren, and X. Sun, "CNN-BiLSTM with attention for Twitter sentiment classification," IEEE Access, vol. 9, pp. 90161–90170, 2021.

[15] L. Zhang, S. Wang, and B. Liu, "Hierarchical attention for multi-aspect Twitter sentiment analysis," Information Processing and Management, vol. 58, no. 5, 102662, 2021.

[16] Z. Li, X. Ding, and T. Liu, "Relational graph attention networks for aspect-based sentiment," Neural Networks, vol. 149, pp. 58–66, 2022.

[17] N. Yadav, A. Yadav, and M. Kumar, "Multi-channel CNN with domain-adaptive embeddings for Twitter sentiment," Applied Soft Computing, vol. 116, 108292, 2022.

[18] Sun, L. Huang, and X. Qiu, "Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence," in Proc. AAAI, vol. 34, no. 5, 2020, pp. 8950–8957.

[19] F. Barbieri, J. Camacho-Collados, L. Neves, and L. Espinosa-Anke, "TweetEval: Unified benchmark for tweet classification," in Findings of EMNLP, 2020, pp. 1644–1650.

[20] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in Proc. EMNLP, 2021, pp. 6894–6910.

[21] J. Xu, Z. Chen, and B. Liu, "Multi-task BERT for joint sentiment, stance, and hate speech detection on Twitter," Information Sciences, vol. 572, pp. 458–471, 2021.

[22] Liang, H. Su, L. Gui, E. Cambria, and R. Xu, "Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks," Knowledge-Based Systems, vol. 235, 107643, 2022.

[23] Z. Wu and X. Shi, "Sentiment-aware masked language modeling for Twitter," Knowledge-Based Systems, vol. 241, 108241, 2022.

[24] J. Zhou, J. X. Huang, Q. V. Hu, and L. He, "Dual-channel BERT with metadata fusion for Twitter sentiment," Pattern Recognition, vol. 134, 109152, 2023.

[25] W. Rahman and M. Z. Hossain, "BERT-BiLSTM with thread context for conversational Twitter sentiment," Expert Systems With Applications, vol. 213, 119054, 2023.

[26] Y. Guo, X. Wang, J. Hu, and H. Chen, "Few-shot Twitter sentiment with GPT-4 in-context learning," ACM Transactions on Information Systems, vol. 42, no. 3, pp. 1–28, 2024.

[27] S. Chen and T. Qian, "Multi-granularity contrastive learning for aspect-based Twitter sentiment," Information Fusion, vol. 104, 102192, 2025.

[28] M. Wang, Y. Zhang, H. Li, and J. Liu, "Parameter-efficient fine-tuning of LLaMA for Twitter sentiment under domain shift," Neural Computing and Applications, vol. 37, pp. 4421–4438, 2025.

[29] Y. Liu, L. Zhu, M. Zhao, and X. Chen, "Multimodal BERT-ViT for image-text Twitter sentiment," IEEE Transactions on Affective Computing, vol. 16, no. 2, pp. 789–803, 2025.

[30] Hazarika, S. Poria, and L. P. Morency, "Retrieval-augmented sentiment analysis for emerging Twitter vocabulary," in Proc. NAACL, 2026, pp. 1234–1248.

[31] R. Müller, S. Kornblith, and G. Hinton, "When does label smoothing help?" in Proc. NeurIPS, vol. 32, 2019, pp. 4694–4703.

[32] Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune BERT for text classification?" in Proc. CCL, 2019, pp. 194–206.