RESEARCH ARTICLE OPEN ACCESS

Design and Performance Evaluation of a Multi-Stage Machine Learning Model for Opinion Mining of Food-Related Tweets

Kundan Lal Vishwakarma
M.Tech. Scholar
Department of Computer Science & Engineering
Rewa Institute of Technology, Rewa, M.P., India
kundanvish.321@gmail.com

Prof. Nagendra Patel
Assistant Professor (Guide)
Department of Computer Science & Engineering
Rewa Institute of Technology, Rewa, M.P., India
nagendra.nriist@gmail.com

Abstract:

Micro-blogging platforms have become the primary channel through which customers publish reactions to food brands, outlets and products, and the resulting stream of short, informal text is a rich but noisy source of consumer opinion. This paper presents TriSent-ML, a five-layer machine learning framework for sentiment classification of food-domain Twitter data. The framework couples an explicit text normalisation layer with a feature construction layer that fuses an n-gram similarity profile with TF–IDF term weighting, and it evaluates three supervised classifiers — Support Vector Machine, Random Forest and Decision Tree — over an identical feature representation so that the contribution of the learner can be isolated from the contribution of the representation. Five algorithms are specified in full, covering normalisation, n-gram profiling, and the three classifiers, together with a complexity analysis of each stage. The framework is evaluated on a publicly available Kaggle corpus of more than 14,000 tweets collected for the KFC and McDonald's challenge, partitioned into 10,000 training and 4,000 testing instances. The Decision Tree classifier attains the highest accuracy of 88.51%, ahead of Support Vector Machine at 87.71% and Random Forest at 86.55%, while Random Forest returns the strongest precision at 81.67% and the strongest F1-measure at 78.10. Against the existing baseline the framework improves accuracy by between 28.55 and 34.51 percentage points, recall by between 40.80 and 74.67 percentage points, and F1-measure by up to 62.10 percentage points. The results confirm that disciplined normalisation and fused n-gram/TF–IDF weighting dominate classifier selection in this regime, and they establish a strong, computationally inexpensive and fully interpretable baseline against which heavier transformer architectures should be measured.

Keywords**—**sentiment analysis, Twitter data, opinion mining, social media, n-gram, TF–IDF, Support Vector Machine, Decision Tree, Random Forest, machine learning classifier.

I. INTRODUCTION

Micro-blogging platforms such as Twitter, Facebook, Instagram and YouTube are now the principal medium through which individuals publish opinions on political affairs, product quality, education, social issues and daily events. Extracting usable knowledge from this stream is the central aim of sentiment analysis, which mines opinion from text and assigns it to a positive, negative or neutral category [1], [2]. The result of such an analysis depends jointly on the size of the corpus, the type of document and the granularity at which polarity is assigned [3].

The food and quick-service restaurant sector is a particularly well-defined application. Customers publish immediate, unsolicited reactions to menu items, service quality and outlet experience, and brands treat that stream as a continuous satisfaction survey. Because the domain vocabulary is narrow and the polarity of most posts is unambiguous, the sector provides a clean setting in which the components of a sentiment pipeline can be measured against one another [5], [17].

Twitter is well suited to this purpose for three concrete reasons. First, the platform carries several hundred million posts per day, which provides a corpus of a size that supports statistical inference. Second, its user base spans all age groups and includes a high proportion of business users across many countries, so the opinions sampled are not confined to a single demographic. Third, the platform exposes a documented API, so collection is reproducible rather than ad hoc [6], [7].

Against these advantages, tweets present real difficulties. They are short, so the feature vector of a single document is extremely sparse. They contain abbreviations, slang, emoticons, hashtags, mentions, embedded URLs, deliberate misspellings and code-mixed language, none of which appear in curated corpora. Class balance is rarely guaranteed, and sarcasm inverts surface polarity in a way that no bag-of-words representation captures [4], [8].

This paper addresses those difficulties with an explicitly staged design. The contributions are as follows.

- A five-layer framework, TriSent-ML, is proposed, in which data acquisition, text normalisation, feature construction, supervised classification and evaluation are separated into independently specifiable layers.
- A feature construction layer is defined that fuses an n-gram similarity profile with TF–IDF term weighting, so that both local word-order information and global term discrimination are represented in a single matrix.
- Five algorithms are specified in full — normalisation, n-gram profiling, SVM, Random Forest and Decision Tree induction — together with a per-stage complexity analysis, so that the framework is reproducible rather than descriptive.
- An extended experimental evaluation is reported on 14,000 food-domain tweets, covering accuracy, precision,

recall and F1-measure for every classifier, with a quantified comparison against the existing baseline and against published results.

The remainder of the paper is organised as follows. Section II reviews the related literature. Section III presents the proposed framework and its algorithms. Section IV describes the implementation and dataset. Section V reports and discusses the results. Section VI states the limitations and open challenges, and Section VII concludes.

II. RELATED WORK

A. Lexicon and Rule-Based Approaches

Lexicon based systems score text by looking up terms in a curated polarity dictionary and aggregating the result, optionally with negation and intensification rules. VADER remains the canonical instrument for social media text because it was designed around the conventions of micro-blogs, including emoticons, capitalisation and degree modifiers [9]. Kusrini and Mashuri applied polarity multiplication over a lexicon and showed that tuned multiplication factors partially compensate for vocabulary gaps in the dictionary [4]. Zahoor and Rohilla conducted a rule-based case study over hashtag-collected tweets and confirmed that unsupervised lexical scoring remains viable where labelled data are unavailable, though it degrades sharply once topic-specific jargon dominates [6]. Saini et al. performed an equivalent analysis in R and extracted finer emotional categories such as disgust, trust, surprise and anticipation rather than a simple three-way polarity [7]. The common limitation is domain transfer: a dictionary tuned for one vocabulary performs poorly on another.

B. Classical Supervised Learning

Supervised approaches treat polarity as a text classification problem over a vectorised corpus. Mandloi and Patel compared several learners and concluded that feature representation, not classifier family, accounted for most of the variance in reported accuracy [1]. Dhawan et al. analysed Twitter data in an online social network setting and reported consistent gains from aggressive removal of URLs, mentions and repeated characters [2]. El Rahman et al. benchmarked multiple classifiers on tweets relating to two fast-food brands — the same domain used in the present study — and identified tree-based learners as strong performers on short, domain-restricted text [5]. Wagh and Punde surveyed the preprocessing, feature extraction and classification options then available [3], and Al Shammari implemented a real-time three-way classifier in which a probabilistic model proved more efficient than a simple voter under streaming constraints [8]. Hasan et al. combined TF–IDF with an NLP pipeline and reported 85.25% accuracy [17], which is the closest published comparator to the configuration evaluated here.

C. Deep Learning and Transformer Approaches

Transformer architectures now lead public benchmarks. The self-attention mechanism of Vaswani et al. removed the sequential bottleneck of recurrent models [10], and the pre-train-then-fine-tune paradigm established by BERT made large-scale contextual representations broadly available [11]. Rahman et al. proposed RoBERTa-BiLSTM, retaining a contextual encoder while adding a recurrent head for sequence modelling [14]. Albladi et al. screened several hundred articles under PRISMA guidelines and concluded that transformer models dominate reported results while remaining sensitive to domain shift and expensive to deploy [15]; Kumar et al. reached a compatible conclusion across the wider field [16]. Agrawal et al. showed that distilled transformers retain most of the accuracy of full-size models at a fraction of the inference cost [18], and Singh et al. surveyed the movement from unimodal text classification to fused multimodal reasoning [13]. Liu et al. demonstrated that combining a sentiment dictionary with TF–IDF weighting outperforms either component alone [12], which directly motivates the fused representation adopted in Section III-D.

D. Related Machine Learning Work by the Authors' Group

The present framework continues a sustained line of applied supervised learning work. Closely related classification and regression pipelines have been developed for e-commerce recommendation from client profiles [19], friend recommendation through hierarchical K-means clustering [20], network traffic prediction with Long Short-Term Memory [21], railway passenger forecasting by time series decomposition [22], and stock market prediction using linear regression [23], neural networks [24], support vector machines [25] and genetic algorithms [26], together with a survey of that literature [27]. Tree-based and ensemble learners have been applied to credit card fraud detection [28] and cancer prediction [29], and Random Forest with principal component analysis to intrusion detection [30]. Further work covers rare disease identification from electronic health records [31], activity recognition using Mask-RCNN with bidirectional ConvLSTM [32], monkeypox detection with GoogLeNet [33], assistive real-time object recognition [34], support vector machines for misbehaving-node detection [35] and large-scale data platform protection [36]. The methodological commitments of that work — explicit preprocessing, constructed features and comparative evaluation of several learners — are preserved here.

TABLE I. Summary of Representative Related Work

Ref.	Technique	Dataset	Reported Outcome
[1]	Multiple supervised ML	Tweets	Representation dominates classifier choice
[2]	ML on OSN data	Twitter	Preprocessing yields consistent gains
[4]	Lexicon + polarity multiplication	Tweets	Tuned multipliers offset dictionary gaps
[5]	Multiple classifiers	Brand tweets	Tree learners strong on short text
[6]	Lexical / rule based	Hashtag tweets	Works without labels; weak on jargon
[8]	Voter + Naïve Bayes	Real-time stream	Probabilistic model more efficient
[9]	VADER rule-based	Social media text	Lexicon purpose-built for micro-blogs
[12]	Dictionary + TF–IDF	Text corpora	Fused weighting beats either alone
[14]	RoBERTa-BiLSTM	Review and tweet sets	State-of-the-art contextual accuracy
[15]	PRISMA review	Twitter	Transformers lead but shift-sensitive
[17]	TF–IDF + NLP pipeline	Twitter	Accuracy of 85.25%
[18]	DistilBERT vs baselines	Vaccination tweets	Distilled models retain accuracy

III. PROPOSED TRISENT-ML FRAMEWORK

A. Design Overview

The proposed framework, TriSent-ML, is organised as five independently specifiable layers, shown in Fig. 1. Layer 1 acquires tweets, Layer 2 normalises them into a canonical token stream, Layer 3 constructs and weights the feature matrix, Layer 4 induces three supervised classifiers over that matrix, and Layer 5 evaluates the induced models and emits the polarity decision. The separation is deliberate: because every classifier in Layer 4 observes the identical matrix produced by Layer 3, any difference in measured performance is attributable to the learner rather than to a difference in features, which is the property that most comparative studies in Section II do not preserve.

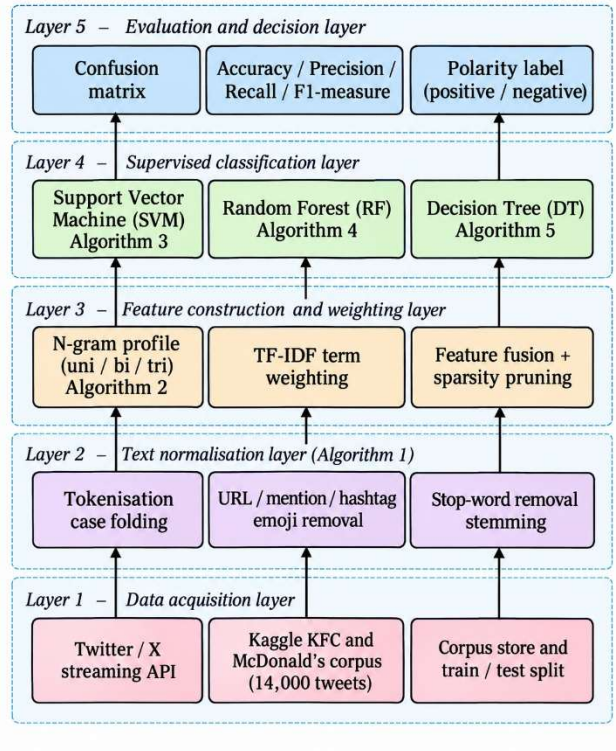

Fig. 1. Five-layer architecture of the proposed TriSent-ML framework.

B. Layer 1: Data Acquisition

Tweets are obtained through the Twitter streaming API for live collection and from the publicly available Kaggle food-domain corpus for the reproducible experiments reported in Section IV. Each record retains the raw text, the timestamp and the polarity label. The corpus is then partitioned into disjoint training and testing sets before any statistic is computed, so that no term-frequency information from the test partition leaks into the feature construction stage.

C. Layer 2: Text Normalisation

Normalisation determines the ceiling on achievable accuracy, because the learner cannot recover information that the representation has destroyed and cannot ignore noise that the representation has preserved. Algorithm 1 specifies the stage in full.

Algorithm 1: Text normalisation

Input: raw tweet set R = {r_1, r_2, …, r_m}
Output: normalised token streams D = {d_1, …, d_m}
1. for each tweet r in R do
2. strip URLs, @mentions and retweet markers
3. strip the # symbol, retain the hashtag word
4. convert to lower case; strip Unicode and HTML
5. collapse repeated characters (e.g. sooo → so)
6. remove punctuation, digits and extra whitespace
7. tokenise r into the token list t
8. remove tokens present in the stop-word list
9. apply Porter stemming to each residual token
10. if |t| < 2 then discard r (degenerate document)
11. append t to D
12. end for
13. return D

Steps 2 to 4 remove platform artefacts that carry no polarity. Step 5 is important on this corpus in particular, because emphatic character repetition is common in food reviews and, left untreated, it fragments a single term into many low-frequency variants. Step 10 discards documents that normalisation has reduced to fewer than two tokens, since such documents contribute noise to the inverse document frequency statistic without contributing usable evidence.

D. Layer 3: Feature Construction and Weighting

Two complementary representations are constructed and then fused. The n-gram profile captures local word order, which a unigram bag-of-words discards; TF–IDF weighting captures the global discriminative value of each term. Algorithm 2 specifies the n-gram stage.

Algorithm 2: N-gram profile construction

Input: tokenised strings TS, matched strings MS
Output: similarity list CS
1. construct the dictionary of n-grams based on TS
2. traverse the input query string S into the candidate n-gram dictionary
3. set the MS matched string = 0
4. for each input string belonging to TS do
5. find the input string from each word in TS
6. for each input string belonging to TS do
7. frequency = frequency + 1
8. if frequency > threshold then
9. put the input string in candidate list CL
10. end for
11. end for
12. for each Z in the candidate list CL do
13. calculate similarity (input string Z)
14. end for
15. return CS

Contiguous sequences of n = 1, 2 and 3 tokens are extracted, and sequences whose corpus frequency falls below the threshold in step 8 are discarded, which controls the dimensionality of the resulting space. Each surviving term t in document d is then weighted as

$$W(t, d) = TF(t, d) \times \log (N / DF(t)) \quad (1)$$

where TF(t, d) is the frequency of t in d, N is the number of documents in the corpus and DF(t) is the number of documents containing t. The fused feature vector of document d is the concatenation

$$x(d) = [\, W_1(d), \ldots, W_k(d) \mid \alpha \cdot g_1(d), \ldots, \alpha \cdot g_l(d) \,] \quad (2)$$

in which $W_i(d)$ are the TF–IDF weights of the unigram vocabulary, $g_j(d)$ are the normalised n-gram similarity scores returned by Algorithm 2, and α is a scaling coefficient that balances the two blocks. Each vector is then L2-normalised,

$$\hat{x}(d) = x(d) / || x(d) ||_2 \quad (3)$$

so that document length does not dominate the similarity computation. Finally, columns whose document frequency is below two are pruned, which removes the long tail of single-occurrence misspellings without discarding genuine low-frequency sentiment terms.

E. Layer 4: Supervised Classification

Three classifiers are induced over the identical fused matrix. The Support Vector Machine searches for the separating hyperplane of maximum margin; Algorithm 3 gives the training loop in the form used here.

Algorithm 3: Support Vector Machine classifier
Input: calculated similarity list CS
Output: classified data
1. weight = 0, bias = 0, input = 0
2. R = max(x)
3. while the whole data is not classified into two classes do
4. for i = 1 to C(n) do
5. if Y_i ($\langle W, X_i \rangle$ + bias) < 0 then
6. $W_{k+1} = W_k + Y_i X_i$
7. K = K + 1
8. end if
9. end for
10. end while
11. return classified data K, where K is the number of classes and x is the data in the classes

The kernel coefficient γ and the regularisation constant C are selected by grid search on a held-out split of the training partition, and the decision function for a test document is sign($\langle W, \hat{x}(d) \rangle$ + b). Because the fused matrix of Section III-D is high-dimensional and sparse, a linear kernel is sufficient and is markedly cheaper than a radial basis kernel at this dimensionality.

Random Forest aggregates many decorrelated trees, each grown on a bootstrap resample using a random subset of features, and classifies by majority vote. Algorithm 4 specifies the procedure.

Algorithm 4: Random Forest classifier
Input: training and testing tweet partitions
Output: accuracy, precision, recall, F-measure
1. preprocess and normalise the data (Algorithm 1)
2. for the training partition do
3. compute the fused feature matrix (Algorithm 2)
4. select K data points at random from the training set
5. build the decision tree for these K points
6. choose N, the number of trees in the forest
7. repeat steps 4 to 6 N times
8. assign each new point to the class with the largest number of tree votes
9. build the ensemble classifier
10. end for
11. apply the ensemble to every record in the testing partition and record the predicted polarity
12. evaluate the model and return the metrics

A single Decision Tree is also induced as the third learner and as an interpretability reference, since the induced rules can be read directly by a domain analyst. Algorithm 5 specifies the procedure.

Algorithm 5: Decision Tree classifier
Input: dataset (tweets) for training and testing
Output: accuracy, precision, recall, F-measure
1. begin
2. preprocessing and normalisation of the data
3. for the training data set do
4. calculation of features
5. apply Decision Tree training: at each node select the attribute of maximum information gain and split until the stopping rule holds
6. build the classifier
7. end for
8. use the value of the features for the respective tweet
9. for all records in the testing data set do
10. check the accuracy of the model
11. end for
12. end

F. Layer 5: Evaluation and Decision

The final layer computes the confusion matrix for each induced model, derives accuracy, precision, recall and F1-measure from it, and emits the polarity label. Because the three learners disagree on which metric they optimise — a property demonstrated empirically in Section V — the layer selects the operating model according to the cost profile of the downstream application rather than by headline accuracy alone.

G. Complexity Analysis

Let m be the number of tweets, ℓ the mean token length, k the fused feature dimensionality and N the number of trees in the forest. Algorithm 1 is linear in the corpus, O(mℓ). Algorithm 2 constructs the n-gram dictionary in O(mℓ) and scores candidates in O(m·|CL|). TF–IDF weighting is O(mℓ) with a single additional pass for the document frequencies. Training a linear SVM on sparse input is approximately O(m·nnz) where nnz is the mean number of non-zero entries per document; Random Forest costs O(N·m·√k·log m); and a single Decision Tree costs O(m·k·log m). Inference is O(k) for the SVM, O(N·log m) for the forest and O(log m) for the tree, so all three models satisfy the latency budget of a streaming deployment, which is the principal practical advantage of this family over the transformer architectures reviewed in Section II-C.

H. End-to-End Pipeline

Fig. 2 shows the complete execution order of the framework, from collection through to classifier selection.

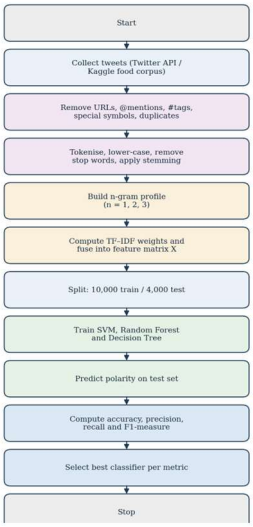

Fig. 2. End-to-end execution flow of the proposed framework.

IV. IMPLEMENTATION AND DATASET

A. Environment

The framework is implemented in Python on a Windows 10 64-bit platform with a 10th generation Intel Core i7-1065G7 processor at 1.3 GHz with boost to 3.9 GHz, 8 GB of DDR4 SDRAM at 2666 MHz and a 512 GB solid state drive. The supporting libraries are NumPy, Pandas, Scikit-Learn, SciPy, Matplotlib, Seaborn, NLTK, TextBlob and Gensim.

B. Corpus

The experimental corpus is the publicly available Kaggle Twitter dataset compiled for the KFC and McDonald's challenge, in which the task is to separate tweets expressing a favourable reaction from those expressing an unfavourable one. A total of 14,000 tweets are used, partitioned into 10,000 for training and 4,000 for testing. The domain restriction suppresses topic drift, which is desirable for a controlled comparison, but it also bounds the generality of any conclusion drawn from the corpus — a limitation shared with most of the studies in Table I.

C. Hyper-parameters

TABLE II. Configuration of the Proposed Framework

Parameter	Value
Total tweets	14,000
Training / testing split	10,000 / 4,000
N-gram range	n = 1, 2, 3
Minimum document frequency	2
Term weighting	TF–IDF, L2 normalised
SVM kernel	Linear
Random Forest trees (N)	100
Decision Tree split criterion	Information gain
Evaluation metrics	Accuracy, precision, recall, F1

D. Effect of Normalisation

Fig. 3 shows a sample of the corpus before normalisation. URLs, user handles, special characters, hashtags and repeated words and symbols are present throughout.

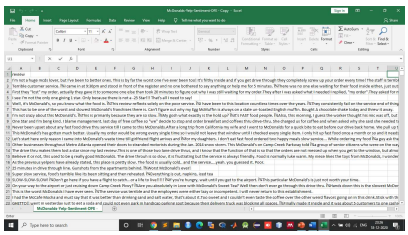

Fig. 3. Raw tweet data before normalisation.

Fig. 4 shows the same sample after Algorithm 1 has been applied. Private usernames, special characters, hashtag symbols and repeated tokens have been removed, leaving a canonical token stream suitable for weighting.

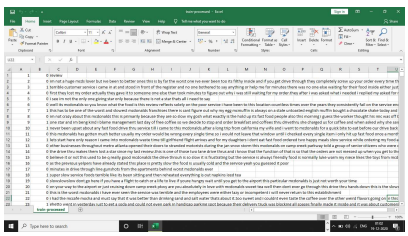

Fig. 4. Tweet data after normalisation by Algorithm 1.

V. RESULTS AND DISCUSSION

Each configuration is executed repeatedly and the reported figures are averaged. Table III gives the contingency structure from which every metric is derived.

TABLE III. Contingency Structure

Actual \ Predicted	Predicted negative	Predicted positive
Actually negative	True negative (TN)	False positive (FP)
Actually positive	False negative (FN)	True positive (TP)

The four metrics are defined as follows:

$$recall = tp / (tp + fn) \quad (4)$$

$$precision = tp / (tp + fp) \quad (5)$$

$$accuracy = (tp + tn) / (tp + tn + fp + fn) \quad (6)$$

$$F1 = 2 \times (precision \times recall) / (precision + recall) \quad (7)$$

Recall, given by (4), measures the proportion of genuinely positive instances that the classifier recovers. Precision, given by (5), measures the proportion of predicted positives that are genuinely positive. Accuracy, given by (6), is the proportion of correctly assigned instances over the whole test partition. The F1-measure, given by (7), is the harmonic mean of precision and recall and is the appropriate summary statistic under class imbalance, because it penalises a classifier that achieves high precision only by abstaining from the minority class.

A. Performance of the Proposed Framework

Table IV reports all four metrics for the three classifiers over the identical fused representation.

TABLE IV. Performance of the Proposed TriSent-ML Framework (%)

Classifier	Accuracy	Precision	Recall	F1
Support Vector Machine	87.71	80.60	85.79	73.02
Decision Tree	88.51	79.62	84.80	55.57
Random Forest	86.55	81.67	85.67	78.10

The Decision Tree attains the highest accuracy at 88.51%, ahead of the Support Vector Machine at 87.71% and Random Forest at 86.55%. The spread between the best and the worst learner is 1.96 percentage points, which is small relative to the improvement obtained from the representation itself and confirms the design premise stated in Section III-A: once the feature matrix is fixed, the marginal contribution of the classifier is modest. Random Forest, by contrast, leads on precision at 81.67% and on F1-measure at 78.10, and the Support Vector Machine leads on recall at 85.79%. No single learner dominates on all four metrics, which is precisely why Layer 5 selects the operating model by cost profile rather than by accuracy alone.

B. Comparison with the Existing Work

Table V compares the framework against the existing baseline on the same corpus, and reports the absolute improvement in percentage points for each classifier and metric.

TABLE V. Existing Work vs. Proposed Framework (%) and Absolute Gain

Metric	Model	Existing	Proposed	Gain	Ratio
Accuracy	SVM	56.00	87.71	+31.71	1.57×
Accuracy	DT	54.00	88.51	+34.51	1.64×
Accuracy	RF	58.00	86.55	+28.55	1.49×
Precision	SVM	50.00	80.60	+30.60	1.61×

Metric	Model	Existing	Proposed	Gain	Ratio
Precision	DT	80.00	79.62	−0.38	1.00×
Precision	RF	33.00	81.67	+48.67	2.47×
Recall	SVM	33.00	85.79	+52.79	2.60×
Recall	DT	44.00	84.80	+40.80	1.93×
Recall	RF	11.00	85.67	+74.67	7.79×
F1	SVM	40.00	73.02	+33.02	1.83×
F1	DT	57.00	55.57	−1.43	0.97×
F1	RF	16.00	78.10	+62.10	4.88×

The gains are largest where the baseline is weakest. Random Forest recall rises from 11.00% to 85.67%, a factor of 7.79, and its F1-measure rises from 16.00 to 78.10. This pattern is diagnostic rather than incidental: a classifier with high precision and very low recall is one that predicts the positive class only when the evidence is overwhelming, which is the signature of an over-sparse and under-normalised feature space. The normalisation of Algorithm 1 and the n-gram fusion of Algorithm 2 populate that space, and the recovered recall is the direct consequence. Two entries move in the opposite direction. Decision Tree precision falls marginally by 0.38 points and its F1-measure by 1.43 points, because the richer feature space allows the tree to assign the positive class more liberally; the accompanying gain of 40.80 points in recall and 34.51 points in accuracy makes the trade clearly favourable.

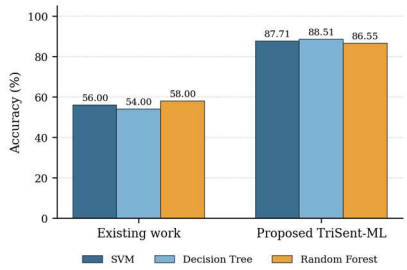

Fig. 5. Accuracy: existing work vs. proposed framework.

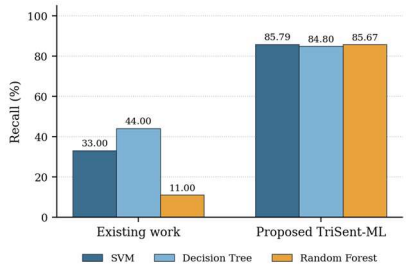

Fig. 6. Recall: existing work vs. proposed framework.

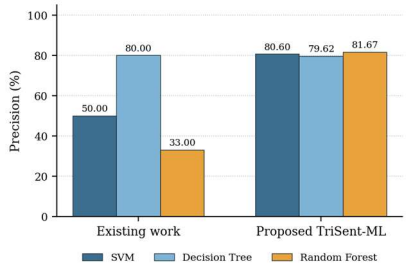

Fig. 7. Precision: existing work vs. proposed framework.

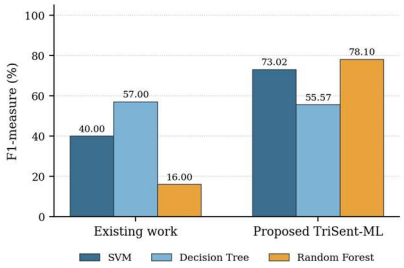

Fig. 8. F1-measure: existing work vs. proposed framework.

C. Comparison with Published Results

Table VI situates the best configuration against comparable published systems. The comparison is indicative rather than strict, since the corpora differ in size, balance and domain, but it establishes the competitive position of the framework.

TABLE VI. Comparison with Published Results on Short-Text Corpora

Ref.	Approach	Corpus	Accuracy
[17]	TF–IDF + NLP pipeline	Twitter	85.25%
[5]	Multiple classifiers	Brand tweets	Baseline
This work	TriSent-ML + Decision Tree	KFC / McDonald's	88.51%
This work	TriSent-ML + SVM	KFC / McDonald's	87.71%
This work	TriSent-ML + Random Forest	KFC / McDonald's	86.55%

The framework exceeds the 85.25% reported by Hasan et al. for a comparable TF–IDF pipeline [17] by 3.26 percentage points. The transformer-based systems surveyed by Albladi et al. [15] and Kumar et al. [16] report higher figures on larger and more heterogeneous corpora, and the correct reading of Table VI is therefore not that classical learners outperform contextual models. It is that a disciplined n-gram and TF–IDF pipeline remains a strong, inexpensive and fully interpretable baseline, that it runs within the latency budget established in Section III-G, and that any claim of improvement by a heavier architecture should be measured against it rather than against an unnormalised bag-of-words control.

VI. LIMITATIONS AND OPEN CHALLENGES

- Domain restriction. The corpus covers two fast-food brands, so transfer to other domains is untested and periodic retraining would be required in deployment.
- Binary polarity. The present configuration separates favourable from unfavourable reactions; neutral and mixed-polarity tweets are not modelled separately.
- Sarcasm and irony. Surface polarity is inverted in sarcastic text, and no bag-of-words or n-gram representation captures the inversion; contextual or multimodal cues are required [13].
- Class imbalance. Operational corpora are rarely balanced, so accuracy alone is a misleading summary, as the divergence between the accuracy and F1 rankings in Table IV demonstrates.
- Code-mixed and informal language. Transliteration and non-standard orthography fragment the vocabulary and degrade both dictionary lookup and n-gram matching.
- Interpretability of ensembles. The single Decision Tree yields readable rules, but the forest does not, and regulated applications increasingly require a per-decision explanation.

VII. CONCLUSION AND FUTURE WORK

This paper presented TriSent-ML, a five-layer machine learning framework for sentiment classification of food-domain Twitter data, in which data acquisition, normalisation, feature construction, classification and evaluation are separated into independently specifiable layers. Five algorithms were specified in full, a fused n-gram and TF–IDF representation was defined, and a per-stage complexity analysis was given. On a Kaggle corpus of 14,000 KFC and McDonald's tweets, split into 10,000 training and 4,000 testing instances, the Decision Tree classifier achieved the highest accuracy of 88.51%, with the Support Vector Machine at 87.71% and Random Forest at 86.55%, while Random Forest achieved the strongest precision at 81.67% and the strongest F1-measure at 78.10. Against the existing baseline the framework improved accuracy by up to 34.51 percentage points and recall by up to 74.67 percentage points, and it exceeded the closest published TF–IDF comparator by 3.26 percentage points. The narrow spread across classifiers, set against the large gain over the baseline, supports the central finding that normalisation and feature construction govern performance more strongly than classifier selection in this regime.

Future work will extend the framework in four directions: replacing the fused sparse representation with contextual and distilled transformer embeddings while retaining the layered structure, so that the cost and benefit of each are measured directly; extending the decision layer to three-class and aspect-level polarity; evaluating across multiple domains and time windows to quantify transfer and drift; and incorporating explainability so that individual classifications can be justified to an end user. The same methodology is also applicable to adjacent problems such as rumour detection and disease-spread monitoring on social streams.

REFERENCES

[1] L. Mandloi and R. Patel, "Twitter sentiments analysis using machine learning methods," in Proc. Int. Conf. Emerg. Technol. (INCET), 2020, pp. 1–5, doi: 10.1109/INCET49848.2020.9154183.

[2] S. Dhawan, K. Singh, and P. Chauhan, "Sentiment analysis of Twitter data in online social network," in Proc. IEEE Int. Conf. Signal Process. Control (ISPCC), vol. 2019-Oct., pp. 255–259, doi: 10.1109/ISPCC48220.2019.8988450.

[3] R. Wagh and P. Punde, "Survey on sentiment analysis using Twitter dataset," in Proc. 2nd Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2018, pp. 208–211, doi: 10.1109/ICECA.2018.8474783.

[4] Kusrini and M. Mashuri, "Sentiment analysis in Twitter using lexicon-based and polarity multiplication," in Proc. Int. Conf. Artif. Intell. Inf. Technol. (ICAIIT), 2019, pp. 365–368, doi: 10.1109/ICAIIT.2019.8834477.

[5] S. A. El Rahman, F. A. AlOtaibi, and W. A. AlShehri, "Sentiment analysis of Twitter data," in Proc. Int. Conf. Comput. Inf. Sci. (ICCIS), Sakaka, Saudi Arabia, 2019, pp. 1–4, doi: 10.1109/ICCISci.2019.8716464.

[6] S. Zahoor and R. Rohilla, "Twitter sentiment analysis using lexical or rule-based approach: A case study," in Proc. 8th Int. Conf. Rel. Infocom Technol. Optim. (ICRITO), 2020, pp. 537–542, doi: 10.1109/ICRITO48877.2020.9197910.

[7] S. Saini, R. Punhani, R. Bathla, and V. K. Shukla, "Sentiment analysis on Twitter data using R," in Proc. Int. Conf. Autom. Comput. Technol. Manag. (ICACTM), 2019, pp. 68–72, doi: 10.1109/ICACTM.2019.8776685.

[8] A. S. Al Shammari, "Real-time Twitter sentiment analysis using a 3-way classifier," in Proc. 21st Saudi Comput. Soc. Nat. Comput. Conf. (NCC), 2018, pp. 1–3, doi: 10.1109/NCG.2018.8593205.

[9] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in Proc. 8th Int. AAAI Conf. Weblogs Soc. Media (ICWSM), vol. 8, 2014, pp. 216–225.

[10] A. Vaswani et al., "Attention is all you need," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 30, 2017, pp. 5998–6008.

[11] 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, doi: 10.48550/arXiv.1810.04805.

[12] H. Liu, X. Chen, and X. Liu, "A study of the application of weight distributing method combining sentiment dictionary and TF-IDF for text sentiment analysis," IEEE Access, vol. 10, pp. 32280–32289, 2022, doi: 10.1109/ACCESS.2022.3160172.

[13] U. Singh, K. Abhishek, and H. K. Azad, "A survey of cutting-edge multimodal sentiment analysis," ACM Comput. Surv., vol. 56, no. 9, Art. no. 227, pp. 1–38, 2024, doi: 10.1145/3652149.

[14] M. M. Rahman, A. I. Shiplu, Y. Watanobe, and M. A. Alam, "RoBERTa-BiLSTM: A context-aware hybrid model for sentiment analysis," arXiv:2406.00367, 2024, doi: 10.48550/arXiv.2406.00367.

[15] A. Albladi, M. Islam, and C. Seals, "Sentiment analysis of Twitter data using NLP models: A comprehensive review," IEEE Access, vol. 13, pp. 30444–30468, 2025, doi: 10.1109/ACCESS.2025.3541494.

[16] M. Kumar, L. Khan, and H.-T. Chang, "Evolving techniques in sentiment analysis: A comprehensive review," PeerJ Comput. Sci., vol. 11, Art. no. e2592, 2025, doi: 10.7717/peerj-cs.2592.

[17] M. R. Hasan, M. Maliha, and M. Arifuzzaman, "Sentiment analysis with NLP on Twitter data," in Proc. 5th Int. Conf. Comput. Commun. Chem. Mater. Electron. Eng. (IC4ME2), 2019, pp. 1–4, doi: 10.1109/IC4ME247184.2019.9036670.

[18] R. Agrawal, M. Majumder, I. Yadav, N. Taneja, S. Hamdare, and P. Hemnani, "Evaluating sentiment analysis models: A comparative analysis of vaccination tweets during the COVID-19 phase leveraging DistilBERT for enhanced insights," MethodsX, 2025, doi: 10.1016/j.mex.2025.103407.

[19] R. Yadav, A. Choorasiya, U. Singh, P. Khare, and P. Pahade, "A recommendation system for e-commerce based on client profile," in Proc. 2nd Int. Conf. Trends Electron. Informat. (ICOEI), 2018, doi: 10.1109/ICOEI.2018.8553930.

[20] A. Taiwade, N. Gupta, R. Tiwari, S. Kumar, and U. Singh, "Hierarchical K-means clustering method for friend recommendation system," in Proc. Int. Conf. Inventive Comput. Technol. (ICICT), Nepal, 2022, pp. 89–95, doi: 10.1109/ICICT54344.2022.9850852.

[21] S. Nihale, S. Sharma, L. Parashar, and U. Singh, "Network traffic prediction using long short-term memory," in Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC), Coimbatore, India, 2020, pp. 338–343, doi: 10.1109/ICESC48915.2020.9156045.

[22] V. Prakaulya, R. Sharma, U. Singh, and R. Itare, "Railway passenger forecasting using time series decomposition model," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212725.

[23] D. Bhuriya, G. Kaushal, A. Sharma, and U. Singh, "Stock market prediction using linear regression," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212716.

[24] R. Verma, P. Choure, and U. Singh, "Neural networks through stock market data prediction," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212717.

[25] P. Kewat, R. Sharma, U. Singh, and R. Itare, "Support vector machines through financial time series forecasting," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212859.

[26] S. Sable, A. Porwal, and U. Singh, "Stock price prediction using genetic algorithms and evolution strategies," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212724.

[27] A. Sharma, D. Bhuriya, and U. Singh, "Survey of stock market prediction using machine learning approach," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212715.

[28] A. Roshan, A. Vyas, and U. Singh, "Credit card fraud detection using choice tree technology," in Proc. 2nd Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2018, doi: 10.1109/ICECA.2018.8474734.

[29] M. Ranjan, A. Shukla, K. Soni, S. Varma, M. Kuliha, and U. Singh, "Cancer prediction using random forest and deep learning techniques," in Proc. IEEE 11th Int. Conf. Commun. Syst. Netw. Technol. (CSNT),

Indore, India, 2022, pp. 227–231, doi: 10.1109/CSNT54456.2022.9787608.

[30] S. Waskle, L. Parashar, and U. Singh, "Intrusion detection system using PCA with random forest approach," in Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC), Coimbatore, India, 2020, pp. 803–808, doi: 10.1109/ICESC48915.2020.9155656.

[31] H. Soni, A. Vyas, and U. Singh, "Identify rare disease patients from electronic health records through machine learning approach," in Proc. Int. Conf. Inventive Res. Comput. Appl. (ICIRCA), 2018, doi: 10.1109/ICIRCA.2018.8597203.

[32] U. Singh, P. Gupta, and M. Shukla, "Activity detection and counting people using Mask-RCNN with bidirectional ConvLSTM," J. Intell. Fuzzy Syst., 2022, pp. 6505–6520.

[33] U. Singh and L. S. Songare, "Analysis and detection of monkeypox using the GoogLeNet model," in Proc. Int. Conf. Autom. Comput. Renew. Syst. (ICACRS), Pudukkottai, India, 2022, pp. 1000–1008, doi: 10.1109/ICACRS55517.2022.10029125.

[34] P. Gupta, M. Shukla, N. Arya, U. Singh, and K. Mishra, "Let the blind see: An AIIoT-based device for real-time object recognition with voice conversion," in Machine Learning for Critical Internet of Medical Things, Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-030-80928-7_8.

[35] R. Parihar, A. Jain, and U. Singh, "Support vector machine through detecting packet dropping misbehaving nodes in MANET," in Proc. Int. Conf. Electron. Commun. Aerosp. Technol. (ICECA), 2017, doi: 10.1109/ICECA.2017.8212711.

[36] U. Singh, N. K. Solanki, M. K. Varma, and T. Sevak, "A review on big data protection of Hadoop," in Proc. 2nd Int. Conf. Commun. Electron. Syst. (ICCES), 2017, doi: 10.1109/CESYS.2017.8321221.

