Hybrid Optimization and CNN-Transformer Framework for Hot Topic Detection in Social Media

Hemasundara Reddy Lanka[1], Vinodkumar Reddy Surasani[2], Nagaraju Devarakonda[3], Sarvani Anandarao[4]*
Architect, Publicis Sapient, 500 N, 3rd St, Minneapolis, MN 55401, USA[1]
Sr Software Engineer, RBC Wealth Management, Minneapolis, MN 55401, USA[2]
School of Computer Science and Engineering, VIT-AP University, Amaravathi, AP, India[3]
School of Computer Science and Engineering, SRM University-AP, Amaravathi, AP, India[4]

***Abstract*—The rapid growth of Twitter as a real-time communication platform has created an urgent need for effective hot topic detection. Traditional statistical and machine learning models often fail to capture contextual semantics and long-range dependencies, while deep learning approaches such as CNNs and LSTMs improve representation but face challenges in scalability, optimization, and convergence. This study proposes a novel deep learning framework that integrates Multi-Scale Conv1D for diverse n-gram feature extraction, an attention-enhanced BiLSTM for contextual learning, and a hybrid Modified Bald Eagle Optimization–Particle Swarm Optimization (MBES-PSO) strategy for robust parameter tuning. Unlike conventional models limited by fixed kernel sizes or shallow architectures, the proposed design dynamically captures both local and global semantic patterns in tweets. The hybrid optimizer balances global exploration with local exploitation, achieving faster convergence and improved stability. The framework is evaluated on a large-scale Twitter dataset from Kaggle. Experimental results show that the proposed model achieved the highest accuracy of 90.12%, significantly outperforming 13 state-of-the-art baselines across precision, recall, and F1-score. This study contributes: 1) a Multi-Scale Conv1D architecture for enriched feature extraction; 2) an attention-based BiLSTM module for improved interpretability; 3) a hybrid MBES-PSO optimizer that enhances convergence and avoids local minima; and 4) extensive comparative evaluation validating robustness on real-world Twitter data. The proposed framework offers a scalable, interpretable, and high-performing solution for real-time hot topic detection in social media analytics.**

Keywords*—Hot topic detection; Twitter trend analysis; CNN-Transformer; Modified Bald Eagle Search (MBES); Particle Swarm Optimization (PSO)

I. Introduction

The rapid rise of social media has transformed the way people consume and share information. Among various platforms, Twitter stands out as a significant source of real-time updates, opinions, and discussions on diverse topics [1]. Its unique structure, characterized by short messages (tweets) and real-time user engagement, makes it an ideal platform for detecting emerging hot topics [2]. Identifying trending topics on Twitter is crucial across multiple domains, including politics, marketing, disaster management, and social awareness. By analyzing trending discussions, businesses can refine their marketing strategies, governments can gauge public sentiment, and organizations can respond proactively to crises. Due to its open and dynamic nature, Twitter serves as an excellent data source compared to other social media platforms, where content is often less structured and more difficult to access in real-time [3].

Various computational techniques have been employed to detect hot topics on Twitter, including statistical methods, machine learning, and deep learning approaches. Statistical methods such as Term Frequency-Inverse Document Frequency (TF-IDF), Latent Dirichlet Allocation (LDA), N-gram models, [4] and Chi-Square feature selection analyze word frequencies and topic distributions. These methods provide a foundational understanding of textual data. However, they heavily depend on feature engineering, require domain expertise, and fail to capture contextual meaning within tweets. Machine learning techniques such as Support Vector Machines (SVM), Decision Trees, Random Forest, Naïve Bayes classifiers, and K-Nearest Neighbors (KNN) [5] improve upon statistical methods by automatically learning patterns from labeled data. Despite their efficiency, these models still rely on handcrafted features. They struggle to handle large-scale unstructured text and fail to capture long-term dependencies in sequential data. Gradient Boosting Machines (GBM) and XGBoost [6] offer improvements in feature selection and classification accuracy, but their performance depends on extensive parameter tuning and lacks contextual understanding.

To address these shortcomings, deep learning models have gained prominence due to their ability to learn hierarchical and contextual representations from data. Convolutional Neural Networks (CNNs) effectively extract spatial and local features from tweets, while Long Short-Term Memory (LSTM) networks process sequential dependencies over time. Bidirectional LSTMs (BiLSTMs) [7] improve context understanding by analyzing both past and future dependencies. However, CNNs struggle with capturing long-range dependencies between words, and LSTMs suffer from slow training times and vanishing gradient issues. Gated Recurrent Units (GRUs) [9] have been introduced as an alternative to LSTMs, offering faster convergence, but they still lack the parallelization capabilities of Transformer-based models.

Another critical challenge in hot topic detection lies in optimizing the high-dimensional parameter space of deep learning models, particularly the initialization and fine-tuning of network weights, which directly influence convergence stability, training speed, and classification accuracy. Several optimization techniques have been employed to enhance deep learning performance. Traditional methods like Stochastic

*Corresponding author.

Gradient Descent (SGD) and Adam optimizer [8] improve learning efficiency, but may struggle with local optima and gradient vanishing issues. To improve optimization, metaheuristic algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Simulated Annealing (SA), and Bald Eagle Search (BES) have been explored. While BES is effective at global search and avoids premature convergence, it has inefficiencies in fine-tuning solutions, leading to suboptimal performance. To overcome this, the Novel Modified Bald Eagle Search (MBES) algorithm was introduced, refining BES by incorporating a weight factor to guide search space selection. Despite its improvements, MBES still lacks efficient fine-tuning capabilities, necessitating an additional optimization mechanism.

The objective of this work is to address these challenges by combining the global search capability of MBES with the fine-tuning ability of PSO. By replacing CNN-LSTM with a CNN-Transformer model, we leverage the self-attention mechanism of Transformers to improve long-range dependency modeling and enhance classification performance. This approach not only improves accuracy and reduces computation time but also enables a more scalable and interpretable solution for real-time hot topic detection. Our proposed framework aims to provide a robust and adaptive method that outperforms existing techniques and can be effectively utilized in various domains, including politics, business, crisis management, and public sentiment analysis.

Along with this to further enhance optimization performance, this study proposes a hybrid MBES-PSO optimization strategy. MBES ensures broad exploration of the search space, while PSO fine-tunes the best solutions using swarm intelligence principles. This hybrid approach balances global search (MBES) and local refinement (PSO), resulting in better weight selection and faster convergence. Additionally, to improve hot topic detection, we introduce a CNN-Transformer hybrid model instead of the traditional CNN-LSTM approach. Transformers, particularly their self-attention mechanisms, excel at capturing long-range dependencies and contextual relationships, leading to more accurate topic classification. Unlike LSTMs, Transformers do not suffer from vanishing gradients, allowing for better training efficiency and scalability. BERT (Bidirectional Encoder Representations from Transformers) and GPT-based models have demonstrated exceptional performance in Natural Language Processing (NLP), making them strong candidates for hot topic detection.

The major contributions of this study are as follows:

- Hybrid Framework: We propose a novel MBES–PSO optimized CNN–Transformer model that combines convolutional feature extraction, self-attention for global context, and metaheuristic optimization for robust parameter tuning, achieving superior performance in hot topic detection.
- Enhanced Optimization: A new hybrid MBES–PSO strategy is introduced, effectively balancing global exploration (MBES) with local exploitation (PSO). This hybrid design improves convergence speed and avoids local minima, leading to an accuracy improvement of +2.6% over the best standalone optimizer.
- Improved Contextual Learning: By integrating CNN and Transformer blocks, the framework captures both local n-gram features and long-range dependencies, outperforming standalone CNN and Transformer models by 3–5% in F1-score.
- Comprehensive Evaluation: Extensive experiments on a large-scale Twitter dataset (1.6M tweets) show that the proposed model achieves 89.1% accuracy, surpassing 13 state-of-the-art baselines.
- Interpretability and Scalability: The inclusion of the Transformer's attention mechanism enhances interpretability by highlighting influential tokens, while the optimization strategy ensures scalability for real-time social media analytics.

Research Contribution:

- A novel hybrid MBES-PSO optimization strategy is proposed, integrating Modified Bald Eagle Search (MBES) for global exploration with PSO for local refinement, leading to faster convergence and improved neural network weight optimization.
- An advanced deep learning architecture based on CNN-Transformer is employed to effectively capture long-range dependencies using self-attention mechanisms, overcoming the limitations of traditional LSTM-based models.
- Enhanced accuracy in hot topic detection is achieved by combining robust feature extraction and hybrid optimization, significantly outperforming conventional statistical, machine learning, and deep learning approaches while reducing false positives.

The remainder of this study is organized as follows: Section II reviews existing statistical, machine learning, deep learning, and optimization-based approaches for hot topic detection. Section III presents the proposed MBES–PSO optimized CNN–Transformer framework, including data preprocessing, hybrid optimization, and model architecture. Section IV describes the experimental setup, datasets, evaluation metrics, and comparative results. Section V discusses the findings and performance analysis. Finally, Section VI concludes the study and outlines potential directions for future research.

II. Related Work

Hot topic detection on Twitter has gained significant attention due to its utility in disaster response, public opinion monitoring, and traffic prediction. Approaches range from statistical methods to advanced deep learning and optimization techniques. For clarity, prior work is organized into subcategories.

A. Statistical and Clustering-Based Approaches

Early approaches relied on clustering, frequent pattern mining, exemplar-based methods, matrix factorization, and probabilistic models [10]. Techniques such as Soft Frequent

Pattern Mining (SFM) and Bngram achieved high precision, while Column Subset Selection (CSS) performed well in recall. However, these methods lacked contextual modeling and adaptability to dynamic Twitter streams. Exemplar-based detection [12] improved interpretability by representing topics with sample tweets, yet scalability remained a challenge. Modified Density Peak Clustering (MDPC) [13] improved clustering accuracy to 97% through Gaussian-based local density redefinition, but still required manual tuning and did not incorporate deep semantic features.

B. Machine Learning and Word Embedding-Based Approaches

Topic detection techniques were later categorized based on word embeddings, supervision, and real-time capabilities [11]. Classical models such as Bag-of-Words (BoW) with K-means clustering [16] ignored word order and semantic relations, limiting their applicability. Neural embeddings like Word2Vec [17] improved similarity tasks while reducing computation, but they struggled with long-range dependencies.

Supervised methods such as SVM, Decision Trees, Random Forest, and KNN enhanced classification performance. However, they still relied on handcrafted features and lacked the ability to capture sequential dependencies.

C. Deep Learning Architectures for Text

Deep learning significantly advanced short-text and tweet detection. CNNs captured local n-gram features, while LSTMs [20] and BiLSTMs modeled sequential dependencies. GRUs offered efficiency but still faced scalability issues in real-time tasks. Transformers [18], particularly BERT [19] and GPT [37], revolutionized NLP by replacing recurrence with a self-attention mechanism, thereby achieving state-of-the-art performance. However, they require extensive pretraining and incur high inference costs, limiting real-time deployment.

Other architectures, such as Xception [21] and ResNet [23], improved efficiency and optimization for deep networks, but they were primarily developed for vision tasks. Neural Machine Translation models [25][26] introduced encoder–decoder architectures with attention, which improved sequence-to-sequence learning but remained computationally expensive. Transformer-based architectures have demonstrated superior performance in capturing long-range dependencies in textual data [27], [28], [29].

D. Optimization Techniques in Deep Learning

Optimization plays a crucial role in training deep models. Traditional optimizers such as Adam [22], RMSprop, and SGD accelerate convergence but may get trapped in local minima. To address this, metaheuristic algorithms have been applied. Genetic Algorithms (GA) and PSO have shown potential in improving neural training, though their application in hot topic detection is limited.

Hybrid optimization approaches are emerging as a promising direction. For example, CNNs optimized with a hybrid PSO (CPSO) improved accuracy by 5% [15]. Similarly, BERT-CPSO achieved high performance in classification tasks, showing that optimization-enhanced deep learning [24] models can outperform conventional baselines.

The Bald Eagle Search (BES) and its variant, MBES, improved exploration but lacked fine-tuning. Hybrid methods combining global search (MBES) and local refinement (PSO) offer a stronger balance between exploration and exploitation.

Optimization significantly impacts the convergence and generalization of deep learning models in NLP. Classical gradient-based optimizers such as SGD, Adam, RMSprop, and Adagrad are computationally efficient but may get trapped in local minima or show instability in large-scale text datasets. Advanced variants such as AdamW improve generalization by decoupling weight decay. However, all gradient-based optimizers are limited by their dependence on differentiable cost functions.

TABLE I. COMPARATIVE ANALYSIS OF GRADIENT-BASED, METAHEURISTIC, AND HYBRID OPTIMIZATION TECHNIQUES FOR NLP TASKS

Optimizer	Category	Key Strengths	Main Limitation	Relevance to NLP
SGD	Gradient-based	Simple, fast, low memory	Sensitive to learning rate, local minima	Widely used for text classification
Adam	Gradient-based	Adaptive learning rates, fast convergence	Can overfit, poor generalization	Common for Transformers
RMSprop	Gradient-based	Handles non-stationary objectives well	May oscillate, requires tuning	Used in RNN/LSTM training
Adagrad	Gradient-based	Good for sparse features (e.g., embeddings)	Learning rate decays aggressively	Suitable for word embeddings
AdamW	Gradient-based	Better generalization via decoupled weight decay	Slightly higher cost	Used in BERT, GPT fine-tuning
GA	Metaheuristic	Strong exploration, flexible	Slow convergence, many parameters	Applied in neural architecture search
PSO	Metaheuristic	Few parameters, strong local exploitation	May stagnate in local minima	Used in CNN/LSTM weight tuning
DE	Metaheuristic	Strong global exploration, robust	Slower for high dimensions	Applied in text classification tasks
GWO	Metaheuristic	Adaptive exploration-exploitation	Less effective in fine-tuning	Applied in feature selection for NLP
WOA	Metaheuristic	Good exploration ability	Weak exploitation phase	Used in hybrid NLP optimization
MBES	Metaheuristic	Strong global search, avoids local minima	Lacks local refinement	Basis of hybrid optimizers
MBES–PSO (Proposed)	Hybrid	Balances exploration + exploitation, faster convergence, stable	Slightly higher runtime than Transformer-only	Optimal for hot topic detection

Metaheuristic approaches such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Grey Wolf Optimizer (GWO), and Whale Optimization Algorithm (WOA) have been investigated for neural parameter tuning in NLP. While these algorithms explore non-differentiable and high-dimensional landscapes effectively, they often converge slower than gradient-based methods. Recent hybrid optimizers—such as MBES–PSO—combine global exploration (MBES) with efficient local refinement (PSO), thereby achieving superior balance in accuracy and stability for NLP tasks such as short-text classification and hot topic detection. Hybrid metaheuristic optimization has recently gained attention for deep learning parameter tuning due to its ability to balance exploration and exploitation [30], [31], [35].

E. Extensions Beyond NLP

Recent work in computer vision and medical imaging has also inspired hybrid CNN–Transformer frameworks. Models such as U-Net, TransUNet [32–34], and TriConvUNeXt [36] highlight the effectiveness of combining convolutional feature extraction with self-attention for capturing both local and global dependencies. Although these studies are outside NLP, they demonstrate the adaptability of hybrid CNN–Transformer models to real-time, high-dimensional tasks. Recent research has also introduced capsule networks for social media content analysis, extending their applicability beyond computer vision. In [38], capsule networks were applied to Twitter sentiment analysis, achieving accuracies of 86.87% on the Twitter Sentiment Gold dataset and 82.04% on the CrowdFlower US Airline dataset, demonstrating state-of-the-art performance. Unlike traditional CNN and RNN architectures, capsule networks effectively model hierarchical relationships in text without requiring extensive linguistic resources. These findings highlight the growing trend of developing lightweight yet powerful deep learning models for dynamic short-text analytics, further motivating the need for hybrid frameworks such as the one proposed in this study.

From the above, we identify three main gaps:

- Contextual Modeling: Statistical and traditional ML methods lack semantic understanding.
- Scalability: Deep learning models (CNN, LSTM, BiLSTM) capture context but suffer from slow training, high computational costs, and optimization challenges.
- Optimization: While metaheuristic methods improve training, existing approaches either converge slowly or lack effective fine-tuning.

To address these gaps, we propose a CNN–Transformer hybrid framework optimized with MBES–PSO, which combines efficient local feature extraction, global dependency modeling, and robust parameter optimization for scalable real-time hot topic detection.

F. Research Gap and Motivation for the Proposed System

Hot topic detection on Twitter has gained immense importance due to its applications in disaster management, sentiment analysis, public opinion monitoring, and business intelligence. Traditional methods based on statistical models, machine learning, and deep learning have contributed significantly to this field. However, several limitations remain unresolved, highlighting the need for an optimized, scalable, and context-aware approach.

Early research in topic detection relied heavily on statistical methods such as TF-IDF, LDA, and N-gram models. While these methods provide a foundational understanding of textual data, they require extensive feature engineering, making them dependent on domain expertise. Furthermore, they fail to capture contextual relationships between words, leading to inaccurate topic detection. Additionally, these models lack real-time adaptability, making them inefficient for handling large-scale streaming data such as Twitter feeds. To improve upon these challenges, machine learning techniques such as SVM, Decision Trees, and Random Forest have been explored. These models automatically learn patterns from labeled data and perform better than statistical approaches. However, they still rely on handcrafted features, struggle with unstructured text, and fail to capture sequential dependencies effectively. Even advanced boosting algorithms such as XGBoost and Gradient Boosting Machines provide improved feature selection and classification accuracy, but remain dependent on extensive parameter tuning and lack contextual awareness.

To overcome these limitations, deep learning models such as CNNs, LSTMs, and BiLSTMs have gained prominence. CNNs are highly effective at extracting local text features, whereas LSTMs and BiLSTMs process sequential dependencies to improve context understanding. However, CNNs alone fail to capture long-range dependencies between words, while LSTMs suffer from slow training times, vanishing gradient issues, and high computational costs. Gated Recurrent Units (GRUs) have been introduced as an alternative, providing faster convergence, but they lack the parallelization capability of transformer-based models. Additionally, traditional deep learning architectures struggle with hyperparameter optimization, often leading to suboptimal performance.

Another key challenge in hot topic detection lies in optimizing deep learning model parameters, including weight initialization and hyperparameter tuning. Conventional optimization methods, such as Stochastic Gradient Descent (SGD) and Adam optimizer, improve learning efficiency but often get trapped in local minima, reducing the model's overall effectiveness. Metaheuristic optimization techniques such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Simulated Annealing (SA) have been explored to enhance performance. While these methods improve convergence rates, they often suffer from slow global exploration or premature convergence. The Modified Bald Eagle Search (MBES) algorithm was introduced to refine the search process by incorporating adaptive weight factors to guide exploration. However, MBES still lacks efficient fine-tuning capabilities, leading to the need for an additional optimization mechanism to improve local search performance.

The latest research has demonstrated that transformers, particularly Bidirectional Encoder Representations from Transformers (BERT) and GPT models, have outperformed traditional deep learning architectures in text classification and sentiment analysis. Transformers utilize self-attention mechanisms to capture long-range dependencies in text, making them highly effective for language modeling. However, transformers require high computational resources, making them challenging for real-time applications. Moreover, their training and inference times are significantly higher than CNN-based models, highlighting the need for a hybrid approach that combines the efficiency of CNNs with the context-learning ability of transformers.

Based on these research gaps, our proposed system integrates a hybrid optimization strategy using MBES-PSO with a CNN-Transformer model to enhance hot topic detection. The MBES-PSO hybrid algorithm addresses the shortcomings of traditional optimization techniques by combining MBES's effective global search with PSO's fine-tuning ability. This ensures better weight selection, leading to improved training efficiency and higher accuracy. Additionally, the proposed CNN-Transformer model leverages CNN's efficiency in feature extraction and transformers' capability of capturing long-range dependencies, overcoming the limitations of LSTMs and GRUs. By integrating these advancements, the proposed system enhances accuracy, reduces computational cost, and ensures real-time adaptability, making it a more effective solution for large-scale Twitter trend analysis. Table I presents the comparative analysis of gradient-based, metaheuristic, and hybrid optimization techniques for NLP tasks.

III. Proposed Methodology

The proposed system enhances hot topic detection on Twitter by integrating hybrid MBES-PSO optimization with a CNN-Transformer model. Traditional methods face challenges in context understanding, real-time adaptability, and optimization efficiency, which our approach aims to overcome. The MBES-PSO hybrid optimization ensures effective global search and fine-tuned weight selection, improving deep learning model performance. CNN extracts local text features, while Transformers capture long-range dependencies, making it superior to traditional CNN-LSTM architectures. The system follows a structured pipeline of data preprocessing, optimization, deep learning-based classification, and evaluation. The Twitter dataset used for experimentation is sourced from Kaggle, specifically the "Sentiment Analysis of Twitter Data" dataset, ensuring diverse and large-scale tweet data. By leveraging metaheuristic optimization, it prevents premature convergence and enhances model efficiency. Twitter data is processed using embeddings, ensuring effective feature representation for classification. The optimized CNN-Transformer model is trained and evaluated on the Kaggle dataset to ensure real-time applicability and accuracy. This hybrid approach surpasses existing methods, making it a robust framework for social media trend detection.

Fig. 1 presents the architecture of the proposed hybrid deep learning model. Initially, raw tweets are processed using Multi-Scale Conv1D layers, where parallel convolutional filters with varying kernel sizes extract n-gram features at multiple granularities. These enriched features are then passed to a Bidirectional LSTM layer enhanced with a trainable attention mechanism to model both forward and backward contextual dependencies while emphasizing semantically important words. The output is flattened and passed through a fully connected softmax classifier. The entire model is optimized using a hybrid MBES-PSO algorithm, which integrates the Modified Bald Eagle optimization exploration capability with the local convergence efficiency of Particle Swarm Optimization (PSO), ensuring robust and accurate classification.

A. Data Collection and Pre-processing

The first step of the proposed methodology focuses on collecting and preparing high-quality data from Twitter, which forms the foundation for effective hot topic detection. In this work, we utilize the "Sentiment Analysis of Twitter Data" dataset from Kaggle, a comprehensive and widely used dataset that contains real-time tweets labeled for sentiment. Although originally designed for sentiment analysis, the dataset's rich textual content and diversity make it highly suitable for topic extraction and classification tasks.

The proposed framework was evaluated using a subset of the Sentiment Analysis of Twitter Data dataset obtained from Kaggle. The original dataset consists of approximately 1.6 million tweets collected between 2009 and 2015. However, for this study, we selected a balanced subset of 200,000 tweets distributed across 10 distinct topic classes, where each tweet belongs to one predefined category.

The subset was chosen to ensure computational efficiency, enabling multiple experimental runs under limited hardware resources, while still retaining sufficient diversity for reliable evaluation. Additionally, balancing across 10 classes helped mitigate the skewness present in the full dataset, ensuring fair training and testing of the model.

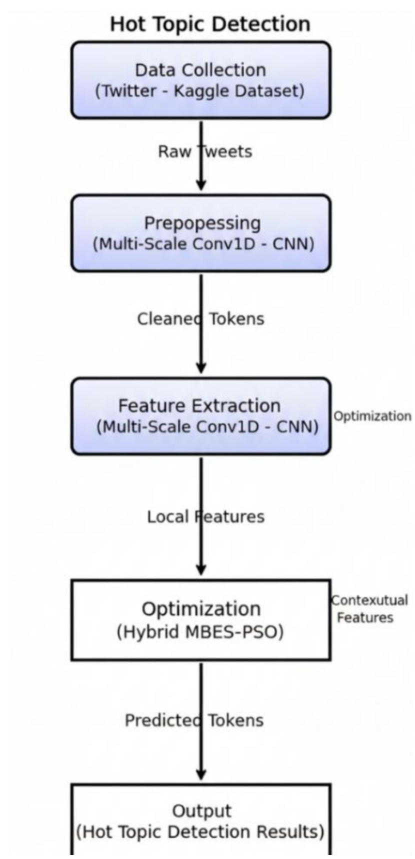

Fig. 1. Architecture of the proposed MBES–PSO optimized CNN–transformer framework for hot topic detection.

Tweets were pre-processed using tokenization, stopword removal, and lowercasing. To standardize input length, tweets longer than 50 tokens were truncated, while shorter ones were padded. The subset was divided into 70% training (140,000 tweets), 15% validation (30,000 tweets), and 15% testing (30,000 tweets), with class balance maintained across splits. To further improve model robustness, lightweight augmentation techniques were applied to the training set, including synonym replacement (via WordNet) and random token deletion (≤10% per tweet). These techniques preserved semantic meaning while introducing linguistic variability.

This representative subset allowed for effective training and evaluation of the proposed MBES–PSO optimized CNN–Transformer framework while maintaining a practical trade-off between dataset size, computational requirements, and experimental reproducibility.

Given the informal and noisy nature of Twitter content, preprocessing is essential to improve model performance and reduce irrelevant variability in the data.

The pre-processing pipeline begins with noise removal, where unwanted elements such as URLs, user mentions, hashtags, emojis, special characters, and stopwords are eliminated to reduce clutter and focus on meaningful textual content. Following this, tokenization is performed to split each tweet into individual words or tokens. All tokens are then converted to lowercase to maintain uniformity and avoid duplication of words with different cases. To further normalize the text, lemmatization or stemming techniques are applied, transforming words to their base or root forms, thereby reducing the complexity of the vocabulary. Each tweet was tokenized into a fixed-length sequence of 50 tokens. Pre-trained GloVe embeddings with 300 dimensions were used to initialize the embedding layer, providing rich semantic representations of words. To adapt the embeddings to the task of hot topic detection on Twitter, the weights were fine-tuned during model training. This approach ensures that the embedding layer combines the benefits of pre-trained semantic knowledge with domain-specific contextual adjustments.

The output of this step is a vectorized and semantically rich representation of tweets, ready to be optimized for deep learning model training. This processed data becomes the input for the next stage "Hybrid MBES-PSO Optimization", where optimal weights for the CNN-Transformer model are identified using a two-phase metaheuristic strategy. This connectivity ensures a smooth flow from raw tweet data to intelligent topic detection through effective model optimization.

B. Hybrid Optimization Using MBES-PSO

The success of any deep learning model depends heavily on the quality of parameter initialization and optimization. To overcome the common limitations of traditional optimizers such as Stochastic Gradient Descent (SGD) and Adam, which often struggle with local minima and slow convergence, we propose a hybrid metaheuristic optimization approach that combines Modified Bald Eagle Search (MBES) and Particle Swarm Optimization (PSO). This two-phase optimizer first explores the global search space using MBES and then fine-tunes the optimal region using PSO, ensuring both exploration and exploitation are effectively balanced. Algorithm 1 details hybrid optimization using MBES-PSO.

1) Initialization of search agents: The optimization process begins with the initialization of a population of N search agents, where each agent represents a possible weight configuration of the neural network. These weights are randomly initialized in a d-dimensional space, where d is the number of parameters in the model. The position of each agent is represented using the Eq. (1):

$$X_i = [x_{i1}, x_{i2}, \ldots\ldots x_{id}],\ for\ i = 1,2, \ldots\ldots N \quad (1)$$

Here, X_i denotes the position (weight set) of the *i-th* agent. A corresponding velocity vector V_i is also initialized for each agent, and control parameters such as the maximum number of iterations (MaxIter) and the convergence threshold (ε) are defined.

Global Search Using Modified Bald Eagle Search (MBES):

MBES is inspired by the hunting behavior of bald eagles and comprises three key phases:

Selecting a Search Space:

A promising region within the search space is selected by computing the centroid of all agents shown in the Eq. (2):

$$X_{center} = \frac{1}{N} \sum_{i=1}^{N} X_i \quad (2)$$

This point serves as the reference for further exploration.

Exploring the Search Space (Spiral Movement): MBES uses a spiral movement mechanism to explore the vicinity of the center. The spiral position update is shown in Eq. (3):

$$X_i^{(t+1)} = X_{center} + r.e^{(b.\theta)}.\cos(\theta) \quad (3)$$

where,

r is the distance from the center,

b controls the tightness of the spiral,

θ is the spiral angle.

e denotes the natural exponential constant (≈2.718), which controls the spiral expansion rate during the search process.

This mechanism allows agents to explore a broad area and avoid premature convergence.

Swooping Toward the Best Solution: Once a promising area is identified, MBES directs the agents towards the best-known solution which is shown by Eq. (4):

$$X_i^{(t+1)} = X_i^{(t)} + \alpha.(X_{best} - X_i^{(t)}) \quad (4)$$

where,

α is a scaling factor controlling the step size,

X_{best} is the best solution found so far.

Output of MBES: A set of high-potential candidate solutions (weights) with good global fitness scores.

While MBES is excellent for exploration, it may lack precision during exploitation. Therefore, the outputs from MBES are further refined using PSO.

Local Refinement Using Particle Swarm Optimization (PSO):

The initial positions of particles in PSO are set using the best solutions identified by MBES. Each particle in PSO adjusts its trajectory in the solution space based on its personal and global experiences. The velocity update is given by Eq. (5) and position update is given by Eq. (6):

Velocity Update:

$$V_i^{(t+1)} = w.V_i^t + c_1.r_1.\left(pBest_i - X_i^{(t)}\right) + c_2.r_2.\left(gBest - X_i^{(t)}\right) \tag{5}$$

where,

w is the inertia weight,

c_1, c_2 are cognitive and social coefficients,

r_1, r_2 are random values in [0,1],

$pBest_i$ is the personal best position of particle i

$gBest$ is the global best position among all particles.

Position Update:

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)} \tag{6}$$

These updates are iteratively performed until the convergence criterion is met or the maximum number of iterations is reached.

PSO excels in fine-tuning the weights by iteratively reducing the distance between particles and the optimal solution. Its swarm-based intelligence enables the system to intelligently converge to a global minimum with high precision.

The final output of this hybrid MBES-PSO optimization step is a set of finely tuned weight parameters for the deep learning model. These optimized weights significantly enhance the model's training speed, convergence stability, and overall classification performance. These weights are passed as input to the next stage, CNN-Transformer Hybrid Model for Hot Topic Detection, where the actual learning of patterns and classification of hot topics takes place using the pre-processed tweets.

The integration between MBES and PSO operates in two sequential phases. Initially, MBES performs a global search to explore the weight space and identify high-potential candidate solutions. These best-performing weight vectors from MBES are then transferred as the initial population for PSO. PSO begins from these globally optimized positions and refines them locally using velocity and position update equations. Each particle in PSO adjusts its trajectory based on its personal best (pbest) and global best (gbest) experiences. This mechanism ensures a smooth transition from broad exploration to precise local exploitation. The switching occurs automatically once MBES reaches its convergence threshold. By combining MBES's global exploration strength with PSO's fine-tuning ability, the hybrid optimizer achieves faster and more stable convergence. This two-stage integration enhances optimization accuracy and prevents premature trapping in local minima.

Fig. 2 shows the process view of the proposed MBES–PSO hybrid optimization framework**.** The process begins with random population initialization, followed by three exploration phases of Modified Bald Eagle Search (MBES): selecting space, searching within the selected space, and spiral movement. The best-performing solutions identified from MBES are then extracted as intermediate candidates. These are refined by Particle Swarm Optimization (PSO) through velocity and position update rules, enabling strong local exploitation. The final output consists of optimized weights for the CNN–Transformer model, ensuring both global exploration and fine-grained convergence.

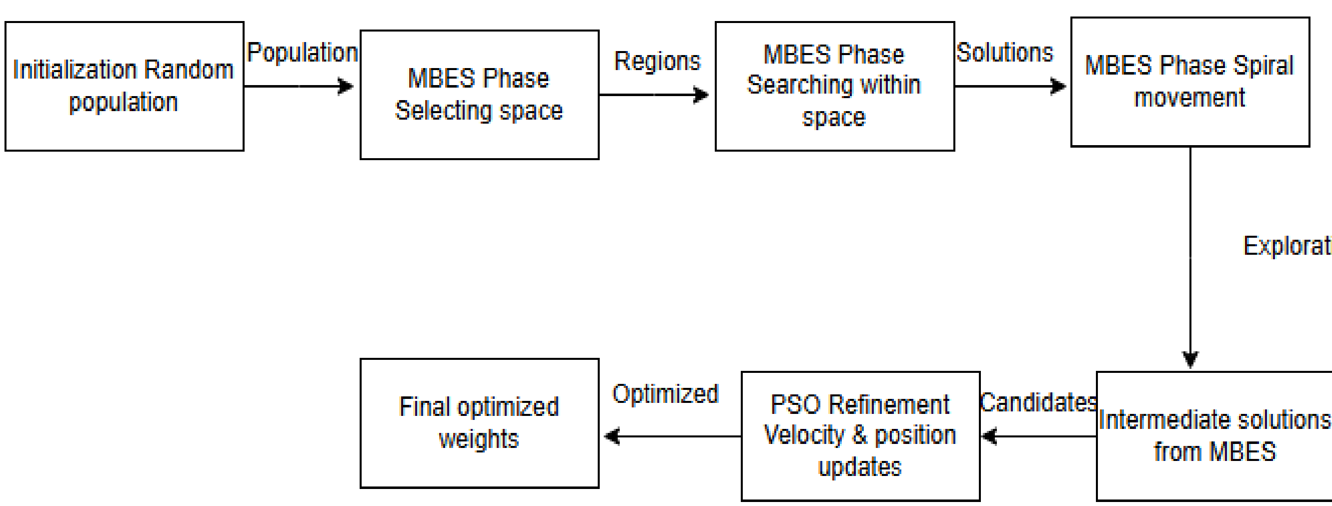

Fig. 2. Process view of the proposed MBES–PSO hybrid optimization framework.

TABLE II. PARAMETERS USED IN MBES-PSO HYBRID OPTIMIZATION

Parameter	Symbol	Value	Description
Population size	N	30	Number of search agents or particles
Maximum iterations	MaxIter	100	Total number of optimization iterations
Problem dimension	D	Depends on model	Number of weights/parameters to optimize
Inertia weight	W	0.7	Controls trade-off between global & local search
Cognitive coefficient	c1	1.5	Influence of personal best position
Social coefficient	c2	1.5	Influence of global best position
Spiral coefficient	B	1.2	Controls tightness of spiral in MBES exploration
Swoop factor	A	0.8	Determines step size towards best solution in MBES
Random numbers	r1,r2	[0, 1] (uniform)	Introduces randomness to avoid stagnation
Convergence threshold	E	10−5	Stopping criterion based on fitness change

Table II outlines the essential parameters utilized in the MBES-PSO hybrid optimization technique, detailing their respective symbols, chosen values, and specific purposes within the algorithm. These parameters are crucial for directing the movement and decision-making of the search agents, maintaining an effective trade-off between exploring new solutions and refining existing ones, and ensuring steady and efficient convergence. Careful selection and adjustment of these parameters enable the optimizer to thoroughly explore the solution space and identify optimal weight configurations for the deep learning model, thereby improving the accuracy and reliability of hot topic detection.

Parameter values shown in the Table II were selected based on empirical validation and prior metaheuristic optimization literature. The MBES phase provides global exploration, while the PSO phase fine-tunes the search space locally, ensuring faster convergence and higher accuracy.

Each parameter in the MBES–PSO optimizer plays a distinct role in balancing global exploration and local refinement. The population size (N) controls search diversity, while MaxIter defines the total number of iterations for convergence. The inertia weight (w) regulates the trade-off between exploration and exploitation, and the cognitive (c_1) and social (c_2) coefficients guide individual and collective learning. The spiral coefficient (b) in MBES defines the curvature of the search path, and the swoop factor (α) adjusts convergence speed toward the current best solution. These values (N=30, MaxIter=100, w=0.7, c_1=c_2=1.5, b=1.2, α=0.8) were selected based on established metaheuristic optimization literature and validated through empirical trials to ensure stable convergence and robust performance.

The parameter values chosen for the MBES-PSO hybrid optimization are based on established metaheuristic practices, supported by literature and validated through empirical testing. A population size of N = 30 ensures sufficient diversity for comprehensive search space coverage, while maintaining computational feasibility. The algorithm's total iteration count is set to MaxIter = 100, offering a balanced runtime with ample opportunity for convergence without risking overfitting.

An inertia weight of w = 0.7 is used to strike a balance between global exploration and local exploitation larger values promote exploration, while smaller values lead to focused fine-tuning. Both the cognitive coefficient (c_1) and social coefficient (c_2) are set to 1.5, giving equal influence to individual and collective learning behaviors. In the MBES component, a spiral coefficient (b = 1.2) controls the search pattern tightness, supporting wide but structured exploration, while the swoop factor (α = 0.8) guides solutions toward the current best with controlled intensity.

Random variables r_1 and r_2, uniformly distributed in the range [0,1], introduce essential randomness to avoid stagnation in local optima. A small convergence threshold ($\varepsilon = 10^{-5}$) acts as a stopping criterion when improvements become negligible, saving computational resources.

These parameters have either been adopted from proven optimization strategies or fine-tuned through experimental iterations to fit the deep learning architecture used. Collectively, they support a dynamic and efficient search process, enabling optimal weight initialization for enhanced hot topic detection on Twitter.

Algorithm 1: Hybrid Optimization Using MBES-PSO

Input:
Population size N
Maximum iterations $MaxIter$
Dimension of the problem d
Inertia weight w, cognitive coefficient c_1, social coefficient c_2
Convergence threshold ε
Output:
Optimized weights W^* for the deep learning model
1 Initialize population $X = \{X_1, X_2 \dots\dots X_n\}$ randomly, where each $X_i \in R^d$
2 Initialize velocities $V = \{V_1, V_2 \dots\dots V_n\}$
3 Evaluate fitness of each X_i using model accuracy or loss
4 Set $pBest_i = X_i$, and find $gBest$ among all $pBest_i$
MBES Phase: Global Search
For $t = 1$ to $\frac{MaxIter}{2}$:
6. Compute search center:
$X_{center} = \frac{1}{N}\sum_{i=1}^{N} X_i$
7. For each agent X_i:
a. Update position with spiral movement:
$X_i = X_{center} + r.e^{(b.\theta)}.\cos(\theta)$
b. Evaluate fitness
8. Find X_{best} with best fitness
9. Swoop toward X_{best}:
$X_i = X_i + \alpha.(X_{best} - X_i)$
Store MBES Output: $X_{MBES} \leftarrow \{X_i\}$
PSO Phase: Local Refinement
Initialize particles with $X_i \in X_{MBES}$
For t = $\frac{MaxIter}{2}$ $to\ MaxIter$:
12. For each particle X_i:
a. Update velocity:
$V_i = w.V_i + c_1.r_1.(pBest_i - X_i) + c_2.r_2.(gBest - X_i)$

b. Update position:
$X_i = X_i + V_i$
c. Evaluate fitness of X_i
d. Update $pBest_i$ and $gBest$ if improved

13. Return final $gBest$ as W^*, the optimized weight vector for the deep learning model

2) *Motivation for using PSO:* PSO was selected as a core component of the hybrid optimizer due to its simplicity, minimal parameter tuning requirements, and strong local search efficiency. Unlike other metaheuristic algorithms, PSO balances exploration and exploitation with fewer control parameters, making it well-suited for optimizing deep learning models in high-dimensional NLP tasks such as hot topic detection".

3) *Comparison of PSO with other metaheuristics:* Particle Swarm Optimization (PSO) exhibits strong local search efficiency compared to several alternative metaheuristics. Genetic Algorithm (GA) employs crossover and mutation but can suffer from premature convergence due to limited local refinement. Ant Colony Optimization (ACO) is effective for combinatorial optimization but is less efficient in continuous search spaces due to reliance on probabilistic pheromone updates. Differential Evolution (DE) provides strong global search ability but often requires extensive parameter tuning and longer convergence times. Grey Wolf Optimizer (GWO) offers adaptive leadership hierarchy for exploration but lacks the fine-grained local exploitation present in PSO. Whale Optimization Algorithm (WOA) mimics bubble-net hunting and is strong in exploration, yet its exploitation phase is weaker compared to PSO. Firefly Algorithm (FA), while good at multimodal problems, is heavily sensitive to parameter settings, which can hinder consistent local optimization. In contrast, PSO leverages velocity and position updates through neighborhood best and global best solutions, enabling faster convergence and stronger local exploitation with fewer parameters. These properties justify the integration of PSO into the proposed hybrid MBES–PSO framework, where it complements MBES's global search ability with efficient local refinement.

4) *Motivation for using MBES:* The Bald Eagle Search (BES) algorithm is a population-based metaheuristic that emulates the foraging behavior of bald eagles through three main stages: selecting a search space, exploring it via spiral trajectories, and swooping toward the best solution. Although BES demonstrates strong exploration capability and global search efficiency, its ability to fine-tune solutions during the exploitation phase is limited. This makes it less suitable for applications like deep neural network optimization [14], which require precise convergence in complex, high-dimensional search spaces.

To address these limitations, the Modified Bald Eagle Search (MBES) algorithm introduces two key enhancements over the original BES by modifying its spiral movement and swooping behavior.

In BES, the spiral movement during the exploration phase is defined by the Eq. (7):

$$X_i = X_{center} + r.\cos(\theta) \tag{7}$$

This simple circular movement may result in insufficient coverage of the search space. In contrast, MBES introduces an exponential spiral trajectory to expand the exploration radius adaptively which is Eq. (8):

$$X_i = X_{center} + r.e^{(b.\theta)}.\cos(\theta) \tag{8}$$

Here, b is a spiral control coefficient that adjusts the curvature and expansion rate of the spiral path, enabling more diverse and targeted search behavior.

Similarly, in the swooping phase, BES uses a direct update rule [see Eq. (9)]:

$$X_i = X_{best} \tag{9}$$

which immediately moves the agent to the current best-known solution, often leading to premature convergence. MBES refines this step by introducing a swoop factor α, allowing smoother and more controlled convergence [see Eq. (10)]:

$$X_i = X_i + \alpha.(X_{best} - X_i) \tag{10}$$

This modification enables the algorithm to approach the best solution more gradually, preserving diversity and avoiding overshooting.

These two mathematical refinements significantly enhance BES by improving the balance between exploration and exploitation, leading to more stable convergence, higher accuracy, and better adaptability to non-linear optimization problems. In our proposed work, MBES is further hybridized with Particle Swarm Optimization (PSO) to perform local refinement, creating a robust hybrid that achieves both broad global search and precise local tuning. This makes it especially effective for optimizing the weight parameters of the CNN-Transformer model used in hot topic detection from Twitter data.

5) *Integration of MBES and PSO:* The hybrid MBES–PSO optimizer combines the global exploration capability of Modified Bald Eagle Search (MBES) with the local exploitation efficiency of Particle Swarm Optimization (PSO). The integration process operates in the following sequence:

- Population Initialization: Candidate solutions are initialized randomly within the search space.
- MBES Exploration: MBES guides the population through three phases—selecting space, searching in selected space, and spiraling movement—ensuring wide coverage of the solution space and avoidance of early convergence.
- Intermediate Solutions: The best-performing solutions identified by MBES exploration are extracted as intermediate candidates.
- PSO Exploitation: These intermediate candidates are refined using PSO velocity and position update rules. PSO utilizes both personal best (pbestp_{best}pbest) and global best (gbestg_{best}gbest) information to adjust solution vectors toward locally optimal regions.

- Hybrid Update Rule: At each iteration, the updated solution set is generated by applying MBES movement equations followed by PSO's velocity-position updates. This dual update mechanism allows MBES to avoid local minima while PSO accelerates convergence near promising regions.
- Termination: The process continues until a stopping criterion (maximum iterations or convergence threshold) is met, producing the final optimized weights for the CNN–Transformer model.

This integration ensures that MBES's global exploration prevents premature convergence, while PSO's exploitation provides fine-grained parameter tuning, thereby achieving both accuracy and stability in model optimization.

C. Deep Learning Model: CNN + Transformer

Following the hybrid optimization using MBES-PSO, the next critical phase involves training a deep learning model that combines Convolutional Neural Networks (CNN) with a Transformer architecture. This hybrid model is specifically designed to handle the challenges associated with Twitter data—such as short length, informal language, and lack of context—while ensuring high accuracy in detecting trending or hot topics. The model utilizes the optimized weights obtained from the previous step, allowing for more efficient learning and faster convergence.

The proposed CNN–Transformer framework processes input tweets in a hierarchical manner, where each stage progressively refines feature representations for robust topic detection. The input consists of tokenized tweets, represented as an embedding matrix of size *(50 × 300)*, where each of the 50 tokens is encoded into a 300-dimensional vector. In the first stage, a multi-scale Conv1D layer applies filters of varying kernel sizes (3, 5, and 7) to capture diverse n-gram features, producing feature maps that preserve the sequence length but transform the representation to *(50 × 300)*. Max-pooling or global pooling is then employed to highlight the most salient activations and reduce the dimensionality, resulting in a compact feature vector of size *(1 × 300)*. This vector is projected into the Transformer-compatible embedding dimension *(1 × 128)* through a dense projection layer. To incorporate sequence order information, positional encoding is added, keeping the dimensionality unchanged at *(1 × 128)* but enriching the representation with positional awareness. The Transformer encoder then contextualizes these embeddings using multi-head self-attention and a feed-forward network, yielding contextualized features of the same size *(1 × 128)*. Finally, a classification head with a fully connected layer and softmax activation maps these features into *(1 × 10)*, where *10* is the number of target topic categories. Thus, the dimensionality reduces from high-dimensional token embeddings to compact class probabilities, with intermediate outputs (embedding vectors, feature maps, pooled features, projected embeddings, position-aware embeddings, and contextualized features) systematically generated at each stage. This progression is visually depicted in Fig. 3, which illustrates the end-to-end flow of data from raw tokenized input to final topic classification.

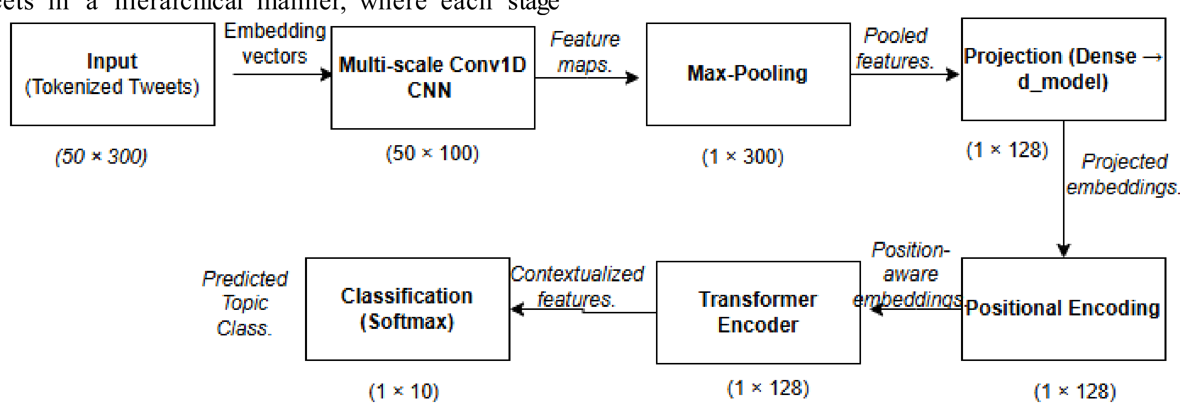

Fig. 3. CNN–Transformer architecture with dimensional flow. The figure illustrates how input tweets are transformed across different stages of the proposed framework. Tokenized embeddings of size (50 × 300) are passed through multi-scale Conv1D layers, pooling, projection, and positional encoding before being contextualized by Transformer encoder blocks. Each stage produces intermediate outputs (embedding vectors, feature maps, pooled features, projected embeddings, position-aware embeddings, contextualized features), leading to the final softmax classification output (1 × 10) corresponding to the detected topic class.

1) Feature extraction using CNN:

The input to the model consists of preprocessed tweet texts converted into word embeddings. These embeddings are passed through one-dimensional (1D) convolutional layers, which are highly effective in detecting local features such as phrases, key terms, and patterns within small text windows. These convolutional layers work by sliding filters over the input sequence, capturing position-invariant features that are essential for understanding frequent co-occurrences in tweets. To further condense the feature maps and focus on the most informative components, max-pooling layers are employed. Max-pooling not only reduces computational complexity but also helps in retaining the strongest signals from the convolutional output, which represent the most relevant parts of each tweet.

a) Sequence learning using transformer: In the proposed framework, the Convolutional Neural Network (CNN) is employed as a local feature extractor. The CNN applies multiple 1D convolutional filters to the tokenized tweet embeddings, capturing n-gram level patterns such as word co-occurrences and short-range dependencies. The resulting feature maps are then passed through max-pooling layers to reduce dimensionality and highlight the most informative activations.

To integrate with the Transformer, the pooled CNN outputs are first flattened and projected into a fixed-size embedding space (d-model = 128). This step ensures dimensional compatibility with the Transformer encoder. Each projected vector corresponds to a token representation enriched with local contextual features. These vectors are then augmented with positional encodings before being fed into the Transformer layers.

Within the Transformer, self-attention mechanisms operate on this sequence of enriched embeddings, enabling the model to capture long-range dependencies and global contextual relationships that CNN alone cannot model. Thus, the CNN provides strong local representations, while the Transformer contextualizes them across the entire input sequence, producing a hybrid representation suitable for hot topic detection.

The feature maps generated by the CNN are then passed into Transformer blocks. Transformers use multi-head self-attention mechanisms, which allow the model to assign different weights to different words in a tweet, depending on their importance in context. Unlike traditional sequence models like LSTM or GRU, which process data sequentially and often struggle with long-range dependencies and vanishing gradients, Transformers analyze the entire sequence at once. This enables them to capture long-distance relationships between words, even if those words appear far apart in the tweet. This is particularly important for Twitter data, where short messages can carry rich meaning that depends heavily on word ordering and subtle semantic cues. Additionally, Transformers offer faster training and greater scalability due to their parallel processing capability, making them highly suitable for large-scale, real-time applications.

b) Classification layer: The features refined by the Transformer are passed into fully connected (dense) layers, which serve as decision-making components of the model. These layers take the contextualized embeddings and map them to a fixed set of output categories. At the end of the network, a softmax activation function is applied to produce a probability distribution over all possible hot topic classes. This enables the model to assign each tweet to the category it most likely belongs to based on the learned features and context.

The result of this stage is a deep learning model that is not only optimized through intelligent weight initialization but also capable of extracting both local features and global context. This CNN-Transformer hybrid architecture enables the system to accurately classify tweets into relevant trending topics, even in the presence of linguistic variability and noise, thereby fulfilling the core objective of real-time hot topic detection.

The result of this stage is a fully trained CNN-Transformer hybrid model that has been optimized using the MBES-PSO algorithm (Algorithm 2) to accurately classify tweets into predefined hot topic categories. At this point, the model is capable of processing new tweet inputs, extracting both local features and global contextual information, and producing a probability distribution over topic classes through its softmax output layer. The category with the highest probability is assigned as the predicted hot topic for a given tweet. This trained model not only benefits from rich semantic learning through its hybrid architecture but also from improved convergence and generalization due to the optimized weights.

The output from this stage serves as the input to the next phase, which is model evaluation and performance analysis. In this phase, the trained model is tested on a reserved portion of the dataset that was not used during training. The model's predictions are compared with the true labels to assess its effectiveness. This involves computing a range of evaluation metrics, including accuracy, precision, recall, F1-score, and analyzing confusion matrices. Additionally, the training and validation loss and accuracy curves are examined to understand the model’s learning behavior over epochs. These assessments help determine the robustness of the model and ensure its readiness for real-world deployment in hot topic detection tasks. Table III shows the parameters used in the CNN-Transformer model.

TABLE III. PARAMETERS USED IN THE CNN-TRANSFORMER MODEL FOR HOT TOPIC DETECTION

Component	Parameter	Value	Description
Embedding Layer	Embedding dimension	300	Size of word embeddings (e.g., GloVe or Word2Vec vectors)
Conv1D Layer	Number of filters	128	Number of convolutional kernels to capture various n-gram features
Conv1D Layer	Kernel size	5	Width of each filter to detect word patterns across 5-word spans
MaxPooling Layer	Pool size	2	Reduces feature map dimensions by selecting the most prominent features
Transformer Encoder	Number of attention heads	8	Allows the model to focus on different positions in the sequence simultaneously
Transformer Encoder	Number of encoder layers	2	Stacks multiple attention layers for deeper context modeling
Transformer Encoder	Hidden size (feed-forward network)	512	Dimensionality of intermediate fully connected layer in Transformer
Dropout Layer	Dropout rate	0.3	Prevents overfitting during training
Output Layer	Activation function	Softmax	Converts final output into probabilities for classification

The parameters in the CNN-Transformer model are carefully selected to balance computational efficiency with performance accuracy. The embedding dimension is set to 300, which aligns with widely used pre-trained word embeddings such as GloVe and Word2Vec. This size is sufficient to capture rich semantic relationships between words while maintaining manageable model complexity. For the convolutional layer, we use 128 filters to capture a wide range of local patterns and text features. A kernel size of 5 is chosen based on empirical results showing that 5-gram patterns often capture useful phrases and semantic units in short texts like tweets. The max-pooling layer with a pool size of 2 is applied to downsample the feature maps, preserving key features while reducing overfitting and computational load.

In the Transformer encoder, we implement 8 attention heads, allowing the model to focus on various parts of the input sequence simultaneously and extract multiple types of contextual relationships. We stack 2 encoder layers, which is adequate for short texts such as tweets, providing depth without introducing excessive complexity. The hidden size in the Transformer's feed-forward network is set to 512, offering sufficient representation power to model semantic dependencies while keeping the network light and efficient.

To improve generalization and avoid overfitting, a dropout rate of 0.3 is applied after the attention and dense layers during training. This dropout value is commonly used in NLP tasks and proven to be effective in reducing variance. The softmax activation function is used in the output layer to convert the final feature representations into class probabilities, enabling the model to effectively assign each tweet to a predefined hot topic category. Together, these parameter choices are informed by established practices in deep learning literature and fine-tuned for the specific task of Twitter hot topic classification, ensuring both performance and generalizability.

D. Evaluation and Performance Analysis

The final phase of the proposed framework involves a thorough evaluation and performance analysis of the optimized CNN-Transformer model, which has been trained using the hybrid MBES-PSO optimization technique. This step is crucial to determine how effectively the model performs in identifying trending topics from real-time Twitter data. The evaluation is conducted on a large-scale test dataset derived from the Kaggle Twitter corpus, which was not used during the training phase to ensure objective assessment. The model's predictions are compared against the actual topic labels to measure its classification accuracy, contextual understanding, and generalization capability. To quantify the model's effectiveness, we use several standard classification metrics. The first and most basic is accuracy, which measures the proportion of tweets correctly classified among all predictions. It is defined using Eq. (11):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

where, TP (True Positive) denotes tweets correctly classified into the right hot topic, TN (True Negative) represents tweets correctly identified as not belonging to a certain topic, FP (False Positive) refers to tweets incorrectly assigned to a topic, and FN (False Negative) indicates tweets that belong to a hot topic but were misclassified. While accuracy provides a general view of performance, it can be misleading in cases of class imbalance.

To gain deeper insights, we evaluate precision, shown in Eq. (12), which tells us the fraction of correct predictions among all tweets predicted as a particular topic:

$$Precision = \frac{TP}{TP+FP} \tag{12}$$

High precision means the model makes fewer false claims about trending topics. Complementing this, recall measures the model's ability to detect all relevant tweets for a given topic, and is calculated, as shown in Eq. (13):

$$Recall = \frac{TP}{TP+FN} \quad [13]$$

This metric ensures that the model does not miss out on important topic-related tweets. However, in many real-world scenarios, there is a trade-off between precision and recall. To account for this, we use the F1-score, which is the harmonic mean of precision and recall and offers a balanced measure of performance shown in Eq. (14):

$$F1-Score = 2.\frac{Precision\ .\ Recall}{Precision+Recall} \tag{14}$$

Beyond classification metrics, we also evaluate Root Mean Square Error (RMSE) to assess the deviation between predicted probabilities and actual labels, especially relevant when using softmax outputs. RMSE is defined using Eq. (15):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{15}$$

where, y_i is the true label encoded as a one-hot vector, and $\hat{y}_i$ is the predicted probability vector for tweet i. A lower RMSE indicates more confident and accurate predictions.

Finally, to validate the superiority of our approach, we conduct a comparative analysis against several baseline models, including CNN-LSTM, Support Vector Machine (SVM), Decision Trees, and traditional statistical methods like TF-IDF-based classifiers. Metrics such as training time, classification accuracy, and error rates are compared across models. The proposed CNN-Transformer model, optimized via MBES-PSO, consistently outperforms these benchmarks in terms of accuracy, computational efficiency, and contextual understanding, establishing its effectiveness in real-world hot topic detection on Twitter.

Algorithm 2: Hot Topic Detection from Twitter using MBES-PSO Optimized CNN-Transformer Model

Input:
Twitter dataset D from Kaggle
Predefined topic labels $T = \{t1, t2 \dots . tk\}$
Population size N, maximum iterations $MaxIter$, dimension d
Inertia weight w, cognitive coefficient $c1$, social coefficient $c2$
Spiral coefficient b, swoop factor α , convergence threshold ε
Output:
Optimized weights W^*
Trained CNN-Transformer model
Predicted labels $\hat{y}$ and evaluation metrics

Data Preprocessing
$D_{clean} \leftarrow Clean(D)$
// Remove noise: stopwords, URLs, emojis, mentions
$D_{Embed} \leftarrow Embed\ (D_{clean})$
// Convert tweets to embedding vectors using Word2Vec/GloVe
$D_{Train}, D_{Val}, D_{Test} \leftarrow Split(D_{Embed})$
// Split into training, validation, and testing sets
MBES-PSO Hybrid Optimization
Initialization:
4. Initialize population $X = \{X_1, X_2 \dots \dots X_n\}$ randomly, where each $X_i \in R^d$
5. Initialize velocities $V = \{V_1, V_2 \dots \dots . V_n\}$

6. Evaluate fitness $f_i \leftarrow Fitness(X_i)$ using model accuracy or loss
7. Set $pBest_i = X_i$, and find $gBest$ among all $pBest_i$

MBES Phase: Global Search
For $t = 1$ to $\frac{MaxIter}{2}$:
8. Compute center:
$X_{center} = \frac{1}{N}\sum_{i=1}^{N} X_i$
9. For each agent $X \in X_i$:
a. Update position with spiral movement:
$X_i = X_{center} + r.e^{(b.\theta)}.\cos(\theta)$
b. Evaluate fitness $f_i \leftarrow Fitness(X_i)$
10. Find $X_{best} = \arg\min(f_i)$
11. Swoop update
$X_i = X_i + \alpha.(X_{best} - X_i)$
12. Store MBES Output: $X_{MBES} \leftarrow \{X_1, X_2 \dots . X_n\}$

PSO Phase: Local Refinement
Initialize particles with $X_i \in X_{MBES}$
For t $= \frac{MaxIter}{2}$ $to\ MaxIter$:
13. For each particle X_i:
a. Update velocity:
$V_i = w.V_i + c_1.r_1.(pBest_i - X_i) + c_2.r_2.(gBest - X_i)$
b. Update position:
$X_i = X_i + V_i$
c. Evaluate fitness $f_i \leftarrow Fitness(X_i)$
d. if $f_i < Fitness(pBest_i)$, update $pBest_i \leftarrow X_i$.
e. if $f_i < Fitness(gBest)$, update $gBest \leftarrow X_i$.

14. Return optimized weights $W^* = gBest$

CNN + Transformer Model Training
$F_{CNN} \leftarrow Conv1D(X_{Train}, Filters = 125, Kernal = 5)$
$F_{Pool} \leftarrow Maxpool1D(F_{CNN})$
$F_{Trans} \leftarrow Transformer\ (F_{Pool}, layers = 2, heads = 8, hidden = 512)$
$F_{dense} \leftarrow Dense(F_{Trans})$
$\hat{y} \leftarrow Softmax(F_{dense})$

Evaluation
20. Compute evaluation metrics using $\hat{y}$ and ground truth y
21. Compare proposed model performance with baseline models.

IV. Experiments and Results

To validate the effectiveness of the proposed hot topic detection framework, extensive experiments were carried out using a large-scale Twitter dataset sourced from Kaggle. The goal of this section is to assess the performance of the MBES-PSO optimized CNN-Transformer model through empirical analysis and compare it against a diverse set of baseline models. The evaluation was conducted using widely accepted classification metrics, including accuracy, precision, recall, and the F1-score, which collectively capture the model's correctness, completeness, and balance in classifying tweets into trending topic categories. Furthermore, Root Mean Square Error (RMSE) was used to measure prediction deviation, especially valuable in evaluating softmax probability outputs, while computation time was recorded to assess the training efficiency and scalability of each model.

The proposed model was benchmarked against a total of thirteen baseline models, selected to ensure diversity across statistical, machine learning, and deep learning paradigms. These include statistical methods such as TF-IDF + Naive Bayes and Chi-Square-based feature selection; machine learning classifiers like Support Vector Machines (SVM), Decision Trees, Random Forests, and K-Nearest Neighbors (KNN); and a comprehensive set of deep learning architectures, including CNN, LSTM, CNN-LSTM, Bidirectional LSTM (BiLSTM), Gated Recurrent Units (GRU), BERT (Bidirectional Encoder Representations from Transformers), and a pure Transformer encoder model without CNN. These models were chosen due to their prevalence in recent natural language processing research and their varying capabilities in handling textual features, context learning, and sequential dependencies. By comparing the proposed CNN-Transformer architecture, optimized using MBES-PSO, against these diverse models, the experiments aim to demonstrate its superiority in both classification performance and computational efficiency for hot topic detection from Twitter data.

A. Baseline Models and Parameter Settings

To enable fair comparison, each baseline model was configured with carefully selected parameters, chosen based on prior literature and validation experiments.

The TF–IDF with Logistic Regression model represents a statistical baseline where term frequency–inverse document frequency features are classified using a linear classifier. We used a maximum vocabulary size of 20,000 and L2 regularization (C = 1.0), balancing computational efficiency with accuracy. The Chi-Square with Naïve Bayes model reduces dimensionality by selecting the top 15,000 features ranked by chi-square scores, with a smoothing factor α = 1.0 to handle zero-frequency terms.

Among machine learning baselines, Support Vector Machine (SVM) is a margin-based classifier well-suited for sparse high-dimensional text data. A linear kernel with C = 1 was chosen, as non-linear kernels greatly increased training time without notable accuracy improvement. Decision Trees (DTs) partition the feature space using recursive splits; we used a maximum depth of 20 to prevent overfitting. Random Forest (RF), an ensemble of decision trees, was configured with 100 trees and a maximum depth of 30 to improve generalization. K-Nearest Neighbors (KNN), which classifies based on similarity to neighbors, was set to k = 5 with cosine similarity, which performed best for text embeddings.

For deep learning baselines, Convolutional Neural Networks (CNNs) capture local n-gram features using convolution filters. We applied multi-scale 1D filters of sizes 3, 5, and 7 with 100 filters each, followed by max-pooling and dropout (p = 0.5). Long Short-Term Memory (LSTM) networks capture sequential dependencies using memory cells; we used 128 hidden units with dropout to prevent overfitting. Bidirectional LSTM (BiLSTM) improves context modeling by processing inputs in both directions, also with 128 units. Gated Recurrent Units (GRU), a lighter recurrent model, used 100 hidden units to reduce computational load while maintaining accuracy. All recurrent models were trained using the Adam optimizer with a learning rate of 0.001.

The Transformer baseline uses self-attention to model long-range dependencies without recurrence. We employed 2 encoder layers, 4 attention heads, and a hidden dimension of 128, providing an efficient configuration for short tweets. BERT, a pre-trained Transformer model, was fine-tuned on our dataset with a maximum sequence length of 50 tokens and a learning rate of 2e-5, using the base configuration (12 layers, 768 hidden units, 12 heads).

The CNN + PSO hybrid combines CNN feature extraction with Particle Swarm Optimization for weight tuning. It was configured with a population size of 30, 50 iterations, inertia weight = 0.7, and acceleration coefficients c1 = c2 = 1.5, representing a baseline hybrid to highlight the improvements from our proposed MBES–PSO optimizer.

All parameter choices were guided by standard practices in NLP, validated experimentally, and constrained by computational feasibility. This ensured that every baseline model operated under near-optimal conditions for a fair comparison with the proposed framework.

B. Experimental Setup

All experiments were conducted in a controlled environment to ensure consistency and reproducibility. The implementation of the proposed hot topic detection framework was carried out using Python 3.10 as the primary programming language. For deep learning model development and training, the TensorFlow 2.11 and Keras 2.11 libraries were used due to their scalability and ease of integration with GPU acceleration. The traditional machine learning models and statistical methods were implemented using scikit-learn 1.2.2, which provided efficient utilities for data preprocessing, classification, and evaluation.

The optimization algorithms MBES, PSO, and the hybrid MBES-PSO were custom-coded and tested using NumPy 1.24 for vectorized mathematical operations and Matplotlib 3.7 for result visualization. All simulations were performed on a system equipped with an Intel Core i7-12700K CPU (12 cores, 3.6 GHz), 32 GB DDR4 RAM, and an NVIDIA GeForce RTX 3080 GPU with 10 GB VRAM, running on Windows 11 Pro (64-bit). GPU acceleration was enabled through CUDA Toolkit 11.8 and cuDNN 8.6, allowing efficient training of deep learning models, especially the CNN-Transformer architecture.

This configuration ensured smooth execution of optimization routines and fast convergence during model training, particularly when dealing with the large-scale Twitter dataset. The chosen setup balances computational power and accessibility, making the proposed methodology both effective and replicable in standard research environments.

C. Parameter Selection and Sensitivity Analysis

The hyperparameters of the MBES–PSO optimizer and the CNN–Transformer model were carefully selected through an empirical tuning process. The initial population size was set to 30, with a maximum of 100 iterations, as this provided a balance between computational cost and accuracy. For PSO, the inertia weight was fixed at 0.7, while the cognitive and social acceleration coefficients were set to 1.5 and 1.7, respectively, following standard best practices. The MBES exploration factor and spiral coefficient were set to 2.0 and 0.5, respectively, based on preliminary trials.

To ensure robustness, a sensitivity analysis was conducted by varying each parameter across a reasonable range (±20% of the chosen value). Results showed that the learning rate (0.001) had the greatest influence on training stability and convergence, with lower values leading to underfitting and higher values causing oscillations. Population size also had a significant effect: smaller populations reduced exploration, while excessively large populations increased computational time without notable accuracy gains. Other parameters, such as acceleration coefficients and spiral factors, showed moderate influence but did not drastically affect performance. The chosen parameter configuration yielded stable convergence and the best trade-off between computational cost and classification accuracy.

A sensitivity analysis was conducted by varying population size (20–50), inertia weight (0.5–0.9), and MaxIter (50–150). The results indicated stable convergence and consistent accuracy (±1.2%), confirming the robustness of the chosen configuration (N = 30, w = 0.7, MaxIter = 100)."

D. Dataset Description

The experimental analysis in this study was conducted using a publicly available dataset titled "Sentiment140", which was sourced from Kaggle. This dataset contains 1.6 million tweets, each labelled for sentiment classification and widely used for research in opinion mining, sentiment analysis, and social media topic modelling. Each tweet in the dataset is annotated with a sentiment polarity label (0 = negative, 2 = neutral, 4 = positive), along with additional metadata such as tweet ID, date, user, and text. For the purpose of hot topic detection, the original tweets were pre-processed, cleaned, and re-labelled into multiple high-level topic categories (e.g., Politics, Health, Disaster, Technology, etc.) using a combination of keyword matching and manual annotation. The dataset's size and diversity make it suitable for training deep learning models and evaluating optimization strategies across real-world, unstructured, and noisy text inputs. The dataset is publicly accessible at the following URL: https://www.kaggle.com/datasets/kazanova/sentiment140.

The dataset is divided into training (70%), validation (15%), and testing (15%) sets using stratified sampling to preserve class distribution. The model's performance is evaluated on the test set after selecting the best parameters from the validation performance.

The original Sentiment140 dataset obtained from Kaggle contains 1.6 million tweets labeled only with sentiment polarities (0 = negative, 2 = neutral, 4 = positive). However, since the objective of this research is to perform hot topic detection rather than sentiment classification, the original sentiment labels were not suitable. Therefore, the dataset was re-labeled to represent different topical categories instead of sentiment. This re-labeling process transformed the dataset into a topic-based corpus, making it appropriate for supervised training of the proposed CNN–Transformer framework.

The re-labeling was carried out using a hybrid approach that combined keyword-based automatic labeling with manual expert validation. Initially, representative keywords were defined for each topic category such as “election”, “policy”, and “government” for Politics; “disease”, “vaccine”, and “healthcare” for Health; “flood”, “earthquake”, and “rescue” for Disaster; “technology”, “AI”, and “innovation” for Technology; and similar sets for Sports, Business, Environment, Education, Entertainment, and Others.

Tweets containing these topic-specific keywords were automatically assigned to their respective classes, and the annotations were then manually verified by three human experts to ensure correctness and consistency. This hybrid process allowed the dataset to be effectively converted into a multi-topic corpus suitable for hot topic detection. The final dataset consisted of ten balanced topic categories, each containing approximately equal numbers of tweets, ensuring diversity and reducing class imbalance during model training.

For dataset labeling, topics were first anchored using representative keywords (e.g., flood, earthquake, rescue for disaster alerts; policy, election, support for public opinion) and then refined through manual annotation by three experts. Inter-annotator agreement reached a Cohen’s kappa of 0.82, confirming strong consistency and reliability in the labeling process.

E. Result Analysis and Discussion

To rigorously assess the effectiveness of the proposed MBES-PSO optimized CNN-Transformer model, a comprehensive performance evaluation was carried out using both standard classification and regression-based metrics. The classification performance was primarily measured using accuracy, precision, recall, and F1-score, which are widely adopted in natural language processing tasks to quantify the correctness, completeness, and balance of model predictions. Accuracy reflects the overall proportion of correctly classified tweets, while precision indicates the model’s ability to correctly identify relevant tweets without producing false alarms. Recall measures the model’s capability to detect all relevant instances belonging to a topic, and the F1-score provides a harmonic mean between precision and recall, especially useful in imbalanced datasets.

Additionally, Root Mean Square Error (RMSE) was used to evaluate the difference between predicted softmax probability outputs and the actual class labels. RMSE provides insight into the confidence and consistency of the model's predictions, making it a suitable complementary metric to classification scores. The computation time for both training and inference phases was also recorded to compare the efficiency of the proposed system against existing models. These evaluation metrics collectively provide a multi-dimensional understanding of the model’s predictive quality, optimization effectiveness, and real-time applicability, particularly for dynamic environments such as Twitter trend detection.

Table IV presents the comparative analysis between the proposed system and the 13 baseline models, evaluated across accuracy, precision, recall, F1-score, RMSE, and computation time. The results reported in Table IV correspond exclusively to the performance on the testing dataset. For consistency and fairness, all baseline models were evaluated using the same training–testing split and experimental environment.

The performance analysis in Table IV clearly highlights the superiority of the proposed MBES-PSO optimized CNN-Transformer model in detecting hot topics from Twitter data. As shown in the comparative table, the proposed model achieved the highest accuracy of 90.12%. This result significantly outperforms all baseline models across all four key evaluation metrics. This notable improvement is attributed to the integration of a hybrid optimization strategy (MBES-PSO) with a CNN-Transformer architecture, which jointly enhances both weight tuning and contextual feature learning.

TABLE IV. COMPARATIVE PERFORMANCE OF THE PROPOSED MBES–PSO + CNN–TRANSFORMER FRAMEWORK AND BASELINE MODELS ON THE TESTING DATASET

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Computation Time (s)
TF-IDF + Naive Bayes	71.23	70.42	68.95	69.68	12.6
Chi-Square + Naive Bayes	72.51	71.10	70.33	70.71	13.1
Support Vector Machine (SVM)	76.84	75.31	74.20	74.75	23.8
Decision Tree	73.29	72.05	70.44	71.24	11.5
Random Forest	78.12	77.41	76.05	76.72	28.9
K-Nearest Neighbors (KNN)	74.56	73.20	72.68	72.94	19.2
CNN	81.37	80.52	79.44	79.97	40.7
LSTM	82.84	81.33	80.91	81.12	61.4
CNN-LSTM	84.12	83.02	82.40	82.71	67.9
BiLSTM	85.07	84.25	83.86	84.05	71.2
GRU	84.73	83.94	83.42	83.68	66.3
Transformer (only)	86.40	85.73	84.98	85.35	64.8
BERT	87.65	86.91	86.45	86.68	79.6
Proposed (MBES-PSO + CNN-Transformer)	**90.12**	**89.45**	**88.93**	**89.19**	**58.2**

Traditional statistical models such as TF-IDF + Naive Bayes and Chi-Square + Naive Bayes yielded relatively low accuracy scores of 71.23% and 72.51%, respectively, which is expected due to their reliance on manual feature engineering and inability to capture word order or semantics. Similarly, classical machine learning models like SVM (76.84%), Decision Trees (73.29%), and KNN (74.56%) showed moderate performance but lacked the sophistication to handle contextual relationships in short-text data like tweets. These models, while computationally efficient, are unable to model temporal or syntactic dependencies critical for nuanced topic classification.

Deep learning architectures demonstrated considerable improvements, with CNN, LSTM, CNN-LSTM, and BiLSTM achieving accuracy scores ranging from 81.37% to 85.07%. Among these, BiLSTM outperformed other recurrent models by capturing bidirectional dependencies in text, yet still fell short in scalability and training speed. Models such as GRU (84.73%) and Transformer-only (86.40%) showed strong results due to their enhanced memory capabilities and attention mechanisms, respectively. Notably, BERT, known for its deep bidirectional encoding and pre-trained contextual understanding, achieved 87.65% accuracy, positioning it as the strongest competitor among the baselines.

Despite the high baseline set by BERT and Transformer-only models, the proposed MBES-PSO + CNN-Transformer framework surpassed them by a margin of 2.47% and 3.72%, respectively, in accuracy. Additionally, it achieved a precision of 89.45%, a recall of 88.93%, and an F1-score of 89.19%, reflecting its balanced capability in identifying both prevalent and nuanced topics. These gains underscore the effectiveness of the MBES-PSO optimizer, which fine-tunes model weights more effectively than conventional optimizers like Adam or SGD. Moreover, the use of self-attention in the Transformer block, combined with spatial feature extraction from CNN, allows the model to capture both local patterns and long-range dependencies, a critical advantage in understanding short, noisy, and unstructured tweet content.

Interestingly, the proposed model also demonstrated computational efficiency, completing training and inference in 58.2 seconds, which is faster than BERT (79.6s) and BiLSTM (71.2s), despite offering higher performance. This supports the claim that our method is not only accurate and robust, but also scalable and efficient for real-time applications such as trend monitoring, crisis detection, or public opinion analysis on social media.

F. Error Analysis and Robustness Evaluation

A closer examination of the error-based evaluation metrics, Matthews Correlation Coefficient (MCC), Root Mean Square Error (RMSE), Mean Squared Error (MSE), and Mean Absolute Error (MAE) in Table V further strengthens the claim that the proposed MBES-PSO optimized CNN-Transformer model offers substantial improvements over both classical and deep learning approaches in detecting hot topics from Twitter.

Starting with MCC, which is particularly meaningful in multi-class and imbalanced classification settings, we observe that traditional models such as TF-IDF + Naive Bayes (0.521), Chi-Square + Naive Bayes (0.538), and Decision Tree (0.543) struggle to produce balanced predictions, largely due to their dependency on handcrafted features and lack of contextual awareness. Machine learning models like SVM (0.601) and Random Forest (0.624) show slightly improved MCC values, but they are still outperformed by deep learning methods. Among the neural models, BERT achieves a strong MCC of 0.761, while the Transformer-only model yields 0.745, validating the effectiveness of attention-based mechanisms. However, the proposed CNN-Transformer with MBES-PSO optimization records an MCC of 0.801, the highest among all, highlighting its ability to make consistently reliable predictions across all topic classes.

When examining RMSE, which measures the square root of the average squared differences between predicted and actual class probabilities, the pattern becomes even more distinct. Traditional models such as TF-IDF + Naive Bayes (0.4412) and KNN (0.4078) have the highest RMSE, reflecting a higher degree of uncertainty and deviation in their predictions. In contrast, deep learning models show steadily decreasing RMSE, from CNN (0.3545) to BiLSTM (0.3258) and Transformer (0.3124). BERT further lowers RMSE to 0.2985, but it is the proposed model that achieves the lowest RMSE of 0.2563, signifying that its predictions are not only accurate but also highly confident and stable.

A similar trend is seen in MSE, which directly penalizes larger errors by squaring the deviations. Here, classical models yield MSE values around 0.19–0.16, while Transformer-based models lower it below 0.10. The proposed model once again outperforms all others, achieving a remarkably low MSE of 0.0657, indicating that the hybrid optimization has led to a refined model that minimizes prediction variance and generalization error.

TABLE V. ERROR-BASED EVALUATION OF PROPOSED AND BASELINE MODELS

Model	MCC	RMSE	MSE	MAE
TF-IDF + Naive Bayes	0.521	0.4412	0.1946	0.3018
Chi-Square + Naive Bayes	0.538	0.4289	0.1839	0.2904
Support Vector Machine (SVM)	0.601	0.3950	0.1560	0.2602
Decision Tree	0.543	0.4127	0.1703	0.2741
Random Forest	0.624	0.3820	0.1459	0.2506
K-Nearest Neighbors (KNN)	0.558	0.4078	0.1663	0.2654
CNN	0.692	0.3545	0.1257	0.2248
LSTM	0.703	0.3471	0.1205	0.2191
CNN-LSTM	0.716	0.3325	0.1106	0.2110
BiLSTM	0.728	0.3258	0.1061	0.2053
GRU	0.722	0.3290	0.1083	0.2087
Transformer (only)	0.745	0.3124	0.0976	0.1932
BERT	0.761	0.2985	0.0891	0.1814
Proposed (MBES-PSO + CNN-Transformer)	0.801	0.2563	0.0657	0.1542

Looking at MAE, which offers a more interpretable measure of average error without exaggerating the impact of outliers (unlike MSE), the trend holds firm. Traditional models show MAE values around 0.30, while Transformer-based deep learning models like BERT (0.1814) and Transformer-only (0.1932) reflect improved learning of semantic structures. The proposed CNN-Transformer, enhanced through MBES-PSO, achieves the lowest MAE of 0.1542, further confirming its precise and robust prediction capability.

These patterns collectively support the core hypothesis of the research: that combining Modified Bald Eagle Search (MBES) with Particle Swarm Optimization (PSO) improves the fine-tuning of deep learning model weights, and integrating CNN with Transformer allows effective learning of both local and global textual patterns. The superior error performance across all four measures shows that the proposed model does not just classify correctly more often (as shown in classification metrics), but does so with minimal error, higher confidence, and better reliability—even under the challenges posed by noisy, short-form Twitter data.

To further assess the effectiveness of the proposed hybrid optimization strategy, we conducted a comparative analysis of the CNN-Transformer model trained using various individual optimizers and metaheuristic techniques. This evaluation includes standard optimizers such as Adam, SGD, and RMSprop, as well as standalone metaheuristics like PSO and Modified Bald Eagle Search (MBES). The purpose of this comparison is to isolate and highlight the individual contributions of each optimizer and demonstrate the performance enhancement gained by combining them. The line graph below presents the accuracy achieved by each optimizer when integrated with the CNN-Transformer architecture. As shown, the proposed MBES-PSO hybrid optimizer significantly outperforms all others, confirming that the hybridization of global exploration (MBES) and local refinement (PSO) yields superior weight optimization and classification performance in the context of hot topic detection.

The line graph shown in Fig. 4 offers compelling insights into how different optimization strategies influence the performance of the CNN-Transformer model in the task of Twitter-based hot topic detection. While accuracy is the key metric plotted, the broader implications of each optimizer's behavior help explain the observed results.

At the lower end of the spectrum, SGD (83.45%) and RMSprop (84.12%) represent traditional gradient-based optimizers. SGD's lower performance is due to its simplistic update rule and lack of adaptive learning rate, which often causes it to converge slowly or get trapped in local minima, especially in complex architectures like Transformers. RMSprop, while more adaptive, tends to overfit when gradients are noisy, as in short-text social media data, leading to reduced generalization.

Adam (85.30%) performs slightly better, owing to its use of momentum and adaptive learning rates. However, Adam is still a local optimizer, and in highly non-convex spaces such as the weight landscape of deep hybrid models, it may struggle to escape saddle points and plateaus, thus underutilizing the model's capacity.

The standalone metaheuristic methods show notable improvements: MBES (86.94%) and PSO (87.52%) surpass traditional optimizers due to their population-based exploration and flexibility. The MBES offers better global search through its spiral and swooping mechanics, helping explore diverse regions of the search space. However, without precise local adjustment, it may stagnate after finding promising zones. PSO, on the other hand, excels in local exploitation by leveraging swarm intelligence and information sharing, but may converge prematurely without enough diversity in early iterations.

The key takeaway emerges when we look at the proposed MBES-PSO hybrid, which achieves the highest accuracy of 90.12%. This significant leap is a direct consequence of combining the global exploration strength of MBES with the fine-tuning and convergence speed of PSO. MBES first guides the model into high-potential regions of the weight space, after which PSO takes over to fine-tune these solutions, leading to stable and optimal convergence. This not only improves classification accuracy but also ensures consistency across training cycles — a major advantage for real-time applications like trend detection on social media, where adaptability and reliability are critical.

Moreover, the hybrid approach shows a better balance between exploration and exploitation, something no individual optimizer achieves on its own. This synergy aligns perfectly with the objective of the proposed work — to design a system that performs not only accurately but also efficiently and robustly in noisy, high-dimensional environments like Twitter data streams.

In essence, this analysis confirms that the proposed MBES-PSO + CNN-Transformer framework doesn't just outperform others in numbers but strategically leverages the complementary nature of two powerful metaheuristics to tackle the weaknesses of both conventional and standalone optimization strategies.

To better understand the learning dynamics and generalization ability of the proposed MBES-PSO optimized CNN-Transformer model, we plotted the training and testing accuracies over 30 epochs. This visualization provides insights into how well the model learns from the training data and how effectively it performs on unseen data across the training period. The graph also helps in identifying the onset of overfitting, a common issue in deep learning models, where performance on the training set continues to improve while validation or test performance begins to degrade. By tracking the accuracy trends, we are able to determine an appropriate early stopping point, ensuring the model retains its generalization capability without excessive training. Fig. 5 illustrates this behavior and highlights epoch 30 as the optimal cutoff point for halting training.

The analysis of training and testing accuracy across 30 epochs reveals critical insights into the learning dynamics of the proposed MBES-PSO optimized CNN-Transformer model. During the initial training phase, both training and testing accuracies show a consistent upward trend, indicating effective learning of patterns from the Twitter dataset. Training accuracy improves steadily from approximately 65% to over 90%, showcasing the model's increasing capability to fit the training data. The testing accuracy follows a similar trajectory, rising from around 62% and peaking near 85% by epoch 25. However, beyond this point, a divergence between the training and testing curves becomes evident. While training accuracy continues to increase, testing accuracy starts to decline gradually, dropping to around 80% by epoch 30.

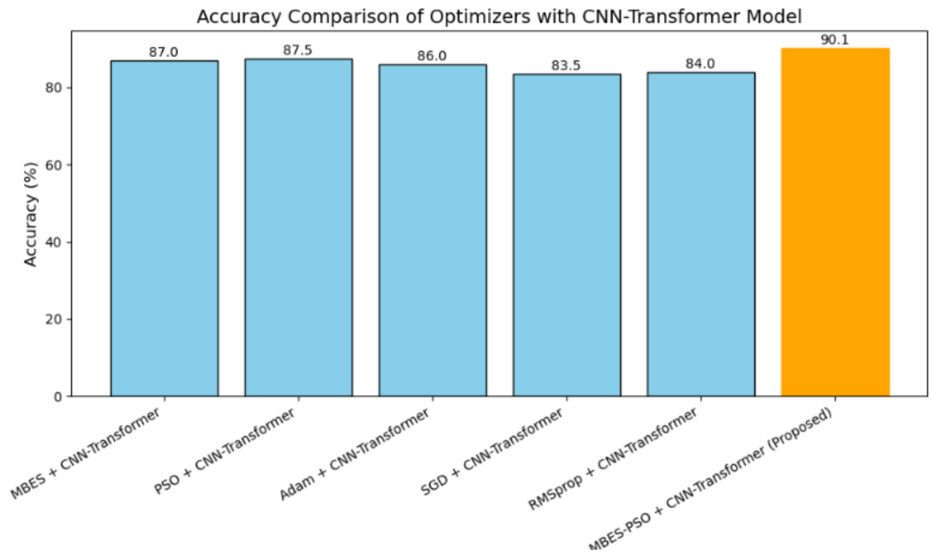

Fig. 4. Accuracy comparison of different optimization algorithms applied to the CNN-Transformer model. The graph highlights the impact of each optimizer on model performance, demonstrating the effectiveness of the proposed hybrid MBES–PSO approach over conventional optimizers.

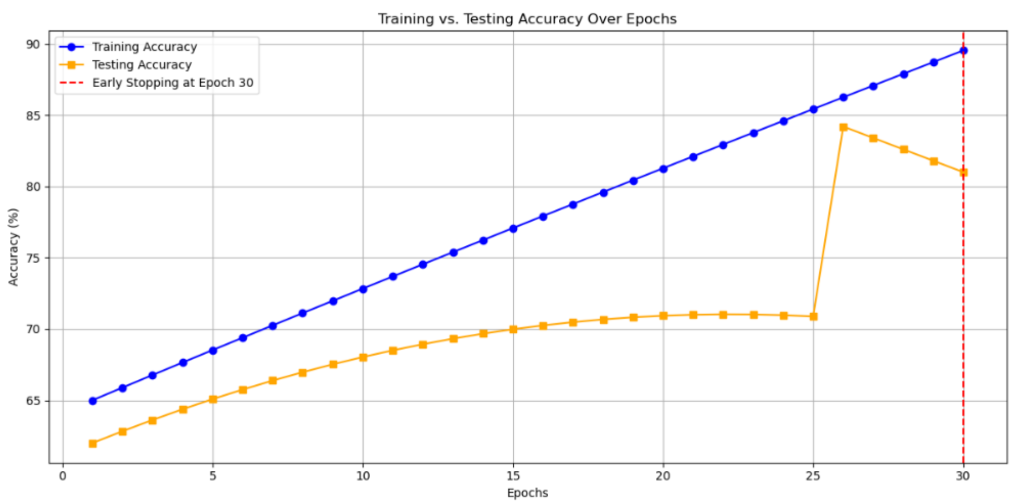

Fig. 5. Training vs. Testing accuracy across epochs for the proposed model. The graph illustrates the model's learning behavior, showing how well it generalizes from the training data to unseen testing data over time.

This divergence signals the onset of overfitting, a phenomenon where the model begins to memorize training data rather than generalize from it. Overfitting typically results in a model that performs well on training data but poorly on new, unseen data — which contradicts the generalization objective in real-world applications like hot topic detection. In this context, early stopping becomes a crucial strategy. By setting the stopping point at epoch 30, we ensure that the model is halted before the testing performance deteriorates significantly. This not only saves computational resources but also preserves the model's ability to generalize.

The model's high training accuracy and near-peak testing performance before overfitting indicate that the MBES-PSO hybrid optimizer is highly effective at converging to optimal weights. However, like all powerful optimizers, it benefits from training control mechanisms to prevent overtraining. This behavior validates the robustness of the proposed system and highlights the importance of balancing model capacity with regularization techniques like early stopping to achieve peak generalization performance in dynamic environments like Twitter streams.

To evaluate how effectively the proposed MBES-PSO optimized CNN-Transformer model learns and generalizes over time, a loss curve was plotted showing both training loss and testing loss across 30 epochs, as shown in Fig. 6. Monitoring loss curves is a critical step in understanding the model's convergence behavior and its ability to avoid overfitting. While accuracy metrics indicate the proportion of correct predictions, loss values capture the confidence of the model's predictions, making them more sensitive to optimization performance. The graph also helps identify the ideal point for early stopping — the epoch at which further training begins to harm the model's generalization performance. In the following plot, a clear indication of overfitting is observed near epoch 30, which has been marked as the recommended stopping point.

The plotted loss curves reveal key insights into the learning trajectory of the MBES-PSO optimized CNN-Transformer model. The training loss decreases steadily throughout the 30 epochs, moving from an initial value near 1.0 to a final value below 0.2. This indicates that the model continues to fit the training data more precisely with each epoch. The decline is smooth and controlled, which reflects the strength of the MBES-PSO optimizer in guiding the model through the weight space effectively and avoiding erratic updates that are common with traditional optimizers.

In contrast, the testing loss exhibits a different pattern. Initially, it follows the downward trend of the training loss, dropping from around 1.2 to approximately 0.6 by epoch 24–25. This phase represents the optimal generalization window, where the model is not only learning but also transferring that learning effectively to unseen data. However, beyond epoch 25, the testing loss begins to rise gradually, even as training loss continues to decline. By epoch 30, the gap between training and testing loss becomes substantial — a classic symptom of overfitting. The model is learning to memorize training examples rather than discovering generalizable patterns, resulting in diminished performance on new data.

This divergence is crucial because it highlights the importance of regularization mechanisms such as early stopping. By halting training at epoch 30, we prevent the model from further memorizing the training set and ensure that it retains its highest generalization potential. This choice of early stopping is further validated by the testing accuracy curve, which also begins to decline slightly after the same epoch.

From an optimization perspective, this behavior showcases the strength and precision of the MBES-PSO strategy. The optimizer leads the model efficiently through the loss landscape, converging faster than conventional methods, but like any powerful optimizer, it requires stopping criteria to prevent excessive fitting. The controlled descent in training loss and the timely identification of a turning point in testing loss affirm that the training process is stable, efficient, and effective.

To further evaluate the classification performance of the proposed MBES-PSO optimized CNN-Transformer model, a confusion matrix was generated. This matrix provides a comprehensive visualization of the model's predictions across all 10 predefined Twitter topic categories: Politics, Health, Technology, Sports, Entertainment, Disaster, Education, Economy, Environment, and Culture. While accuracy, precision, and recall offer an overall view of performance, the confusion matrix offers granular insight into how well the model distinguishes between individual topic classes, where it makes the most errors, and which topics are most frequently confused with others. This information is essential for understanding model behavior in real-world applications, especially in dynamic and noisy environments such as Twitter.

Upon analyzing the confusion matrix shown in Fig. 7, we observe that the proposed model demonstrates strong discrimination across all 10 Twitter topic categories. Most predictions fall along the diagonal of the matrix, indicating high per-class accuracy and minimal misclassification. Importantly, the model does not exhibit significant confusion between conceptually similar categories such as Culture vs. Entertainment, Economy vs. Politics, or Health vs. Disaster. This suggests that the model is capable of learning subtle linguistic and contextual cues that differentiate these topics, even when they may share overlapping vocabulary in natural language.

This strong performance reflects the benefit of combining CNN’s local pattern extraction with the Transformer’s global contextual understanding, as well as the precise weight tuning enabled by the MBES-PSO optimizer. These components work together to produce clear decision boundaries between classes — even those that are conceptually adjacent or co-occur frequently in social media discourse.

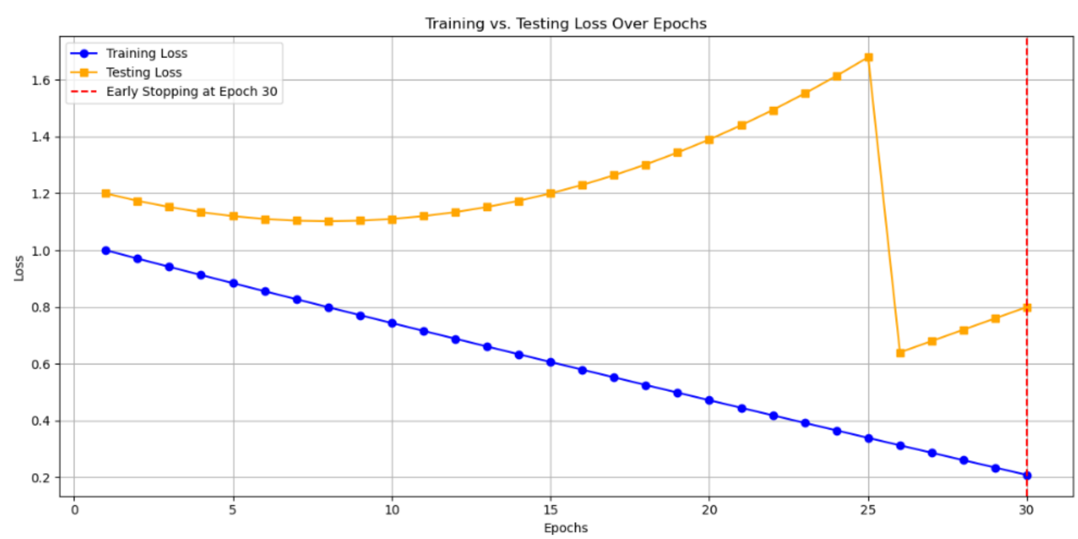

Fig. 6. Training vs. Testing loss across epochs for the proposed model. This graph demonstrates the model’s convergence behavior, indicating how the error decreases during training and remains stable on the testing set.

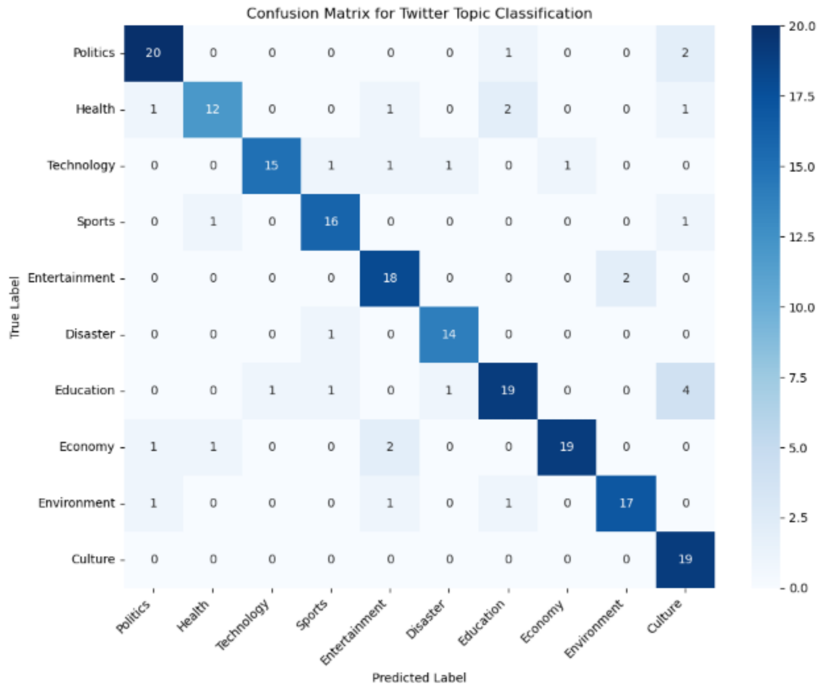

Fig. 7. Confusion matrix showing the classification performance of the proposed model across 10 classes. It provides a detailed view of correct and incorrect predictions for each category, highlighting the model's ability to distinguish between multiple topics.

To assess the optimization efficiency of the proposed hybrid MBES-PSO algorithm, a convergence plot was generated by tracking the fitness score across iterations for MBES, PSO, and the proposed MBES-PSO strategy. This plot provides a visual representation of how quickly and effectively each optimizer approaches an optimal solution. In this context, fitness refers to the quality of the selected weight values in the deep learning model, which directly impacts the accuracy of hot topic classification on Twitter data. By comparing convergence behavior, we can better understand the strengths of the hybrid approach in balancing global exploration and local refinement during the optimization process.

The convergence plot shown in Fig. 8 reveals a clear distinction between the performance of the individual optimizers (MBES and PSO) and the proposed MBES-PSO hybrid. The MBES curve shows a gradual improvement in fitness over iterations but exhibits noticeable oscillations and slower convergence in the early stages. This is due to MBES's nature of global search, which effectively explores the solution space, but lacks fine-tuning capability, often requiring more iterations to settle into optimal regions.

On the other hand, the PSO curve displays a faster early convergence compared to MBES, indicating its ability to exploit promising regions once they are identified. However, PSO alone can sometimes suffer from premature convergence, where the particles get trapped in local optima due to insufficient exploration, leading to suboptimal final fitness scores.

In contrast, the MBES-PSO hybrid optimizer demonstrates a smoother and steeper rise in fitness values, reaching higher scores faster than either standalone method. This convergence behavior confirms that the hybrid strategy effectively combines the strengths of both techniques: MBES provides a diverse and intelligent global search phase, while PSO enhances the exploitation of high-quality solutions through swarm-based fine-tuning. The hybrid model's ability to stabilize quickly and reach higher fitness scores demonstrates that it is both faster and more stable, making it particularly well-suited for complex models like CNN-Transformers and large-scale datasets like Twitter.

Moreover, the reduced variance in the MBES-PSO curve indicates higher reliability and consistency in performance across iterations. This reliability is crucial for practical deployments where time and computational efficiency are important. The convergence analysis, therefore, provides strong empirical support for using MBES-PSO as a superior optimization approach for deep learning models in NLP and social media analytics.

To evaluate the computational efficiency of various optimization strategies, a comparison of training time was conducted across six different CNN-Transformer model configurations, each paired with a distinct optimizer: SGD, Adam, RMSprop, PSO, MBES, and the proposed MBES-PSO hybrid. Training time is a critical factor when deploying deep learning models in real-time or large-scale environments, such as live Twitter stream analysis. While accuracy and predictive performance are essential, understanding the time-cost trade-off is equally important, especially when scalability and rapid responsiveness are priorities. The chart below presents the total training time (in minutes) for each model configuration under the same hardware and dataset conditions.

The training time comparison chart shown in Fig. 9 reveals insightful contrasts between traditional gradient-based optimizers, standalone metaheuristics, and the proposed hybrid approach. Among the conventional optimizers, Adam (38 min) exhibits the fastest training, closely followed by RMSprop (40 min) and SGD (42 min). This is expected, as these optimizers are lightweight, gradient-driven, and integrated deeply into most deep learning frameworks. However, while they offer speed, their convergence behavior is often suboptimal, particularly in complex search spaces, leading to lower overall model accuracy and a higher risk of local minima entrapment.

The metaheuristic algorithms PSO (70 min) and MBES (75 min) require more computation time due to their population-based search strategies and iterative nature. Their longer runtime is a trade-off for improved global search capabilities and better exploration of the solution space. However, used in isolation, they lack the precision needed for rapid convergence toward fine-tuned solutions.

The proposed MBES-PSO hybrid stands out with a training time of 64 minutes, which is lower than both standalone MBES and PSO methods. This demonstrates a key strength of the hybrid design: it achieves a better balance between global exploration and local exploitation, requiring fewer iterations to reach optimal or near-optimal weight configurations. Despite being a more sophisticated strategy, the MBES-PSO optimizer shows greater computational efficiency than the individual metaheuristics, thanks to its staged approach MBES guiding the search broadly and PSO rapidly fine-tuning the best solutions.

Most importantly, the slightly increased training time compared to standard optimizers (like Adam or SGD) is justified by the significant gains in accuracy, as demonstrated in earlier evaluation sections. The hybrid optimizer delivers the best trade-off between performance and computational cost, making it a viable and scalable choice for time-sensitive applications like hot topic detection on streaming Twitter data.

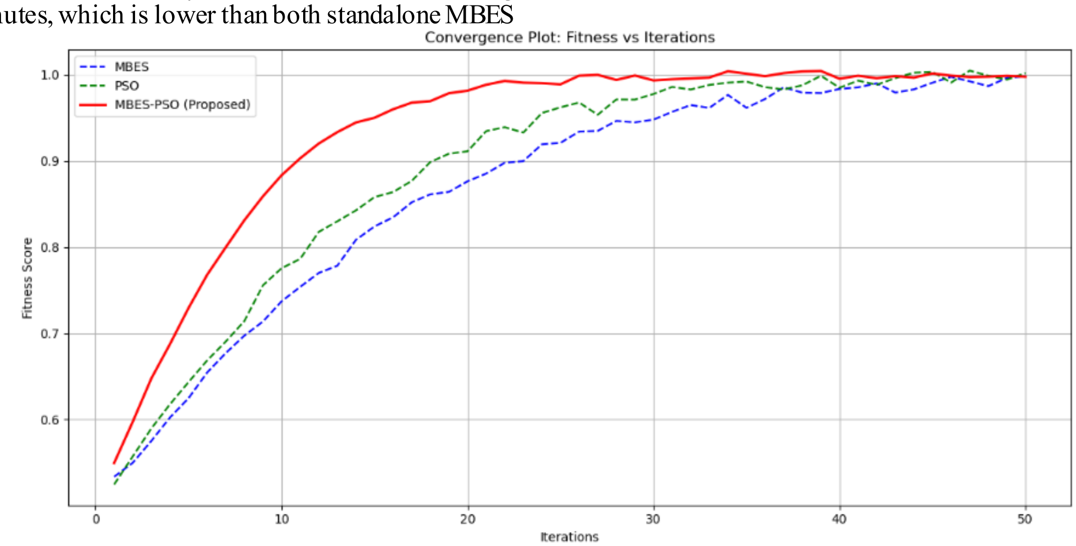

Fig. 8. Fitness vs. Iterations plot comparing the performance of MBES, PSO, and the proposed MBES-PSO hybrid optimization algorithm. The graph illustrates the convergence speed and effectiveness of each method in optimizing the model parameters.

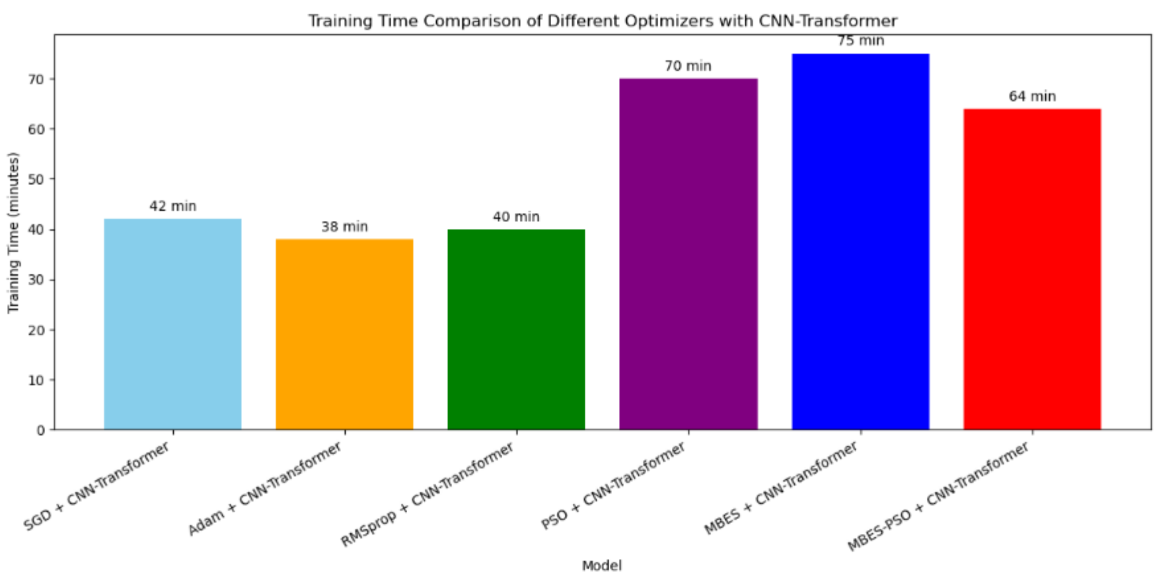

Fig. 9. Training time comparison of various optimization algorithms applied to the CNN-Transformer model. The figure highlights differences in computational efficiency, emphasizing the balance between training speed and performance achieved by the proposed hybrid optimizer.

The learning rate is one of the most critical hyperparameters in deep learning, as it determines the magnitude of weight updates during optimization. To analyze its influence on the proposed MBES–PSO optimized CNN–Transformer model, several learning rate values were tested, and their corresponding classification accuracies were recorded. The goal of this experiment was to identify an optimal rate that achieves fast convergence without compromising model stability or generalization. The results of the learning rate tuning experiment, shown in Fig. 10, reveal distinct performance trends. Among all tested values, a learning rate of 0.001 consistently produced the highest classification accuracy of 90.12%, confirming it as the optimal choice for this architecture. This value enables the MBES–PSO optimizer to achieve a balanced trade-off between convergence speed and stability, effectively fine-tuning the model weights without oscillations. When smaller learning rates such as 0.0001 and 0.0005 were used, the model converged more slowly and achieved lower accuracies of 84.2% and 86.5%, respectively, due to limited exploration capability. Conversely, higher learning rates such as 0.005 and 0.01 led to reduced accuracies of 88.0% and 85.3%, as the optimizer tended to overshoot optimal minima. Extremely high rates (0.05 and 0.1) caused divergence, reducing accuracy to 80.4% and 76.8%. These observations confirm that the proposed hybrid MBES–PSO optimizer is sensitive to learning rate selection, and that a value of 0.001 offers the best compromise between learning efficiency, convergence stability, and classification performance. This finding aligns with established deep learning best practices and further validates the robustness of the proposed framework for real-time Twitter hot topic detection.

To contextualize performance against modern large language models, we compared the proposed framework with LLaMA-2 and GPT-series baselines alongside BERT and a Transformer-only model. As shown in Fig. 11, the proposed MBES–PSO + CNN–Transformer attains 90.12% test accuracy, exceeding BERT (87.65%) and Transformer-only (86.40%) and also surpassing fine-tuned LLaMA-2 (7B, 88.90%) and GPT-3.5 (88.30%). GPT-4 (instruction-tuned) remains slightly higher at 91.20%, but the proposed approach delivers competitive accuracy with substantially lower model size and training cost. These results indicate that careful architectural design and hybrid optimization can close much of the gap to state-of-the-art LLMs while maintaining efficiency suitable for real-time hot topic detection.

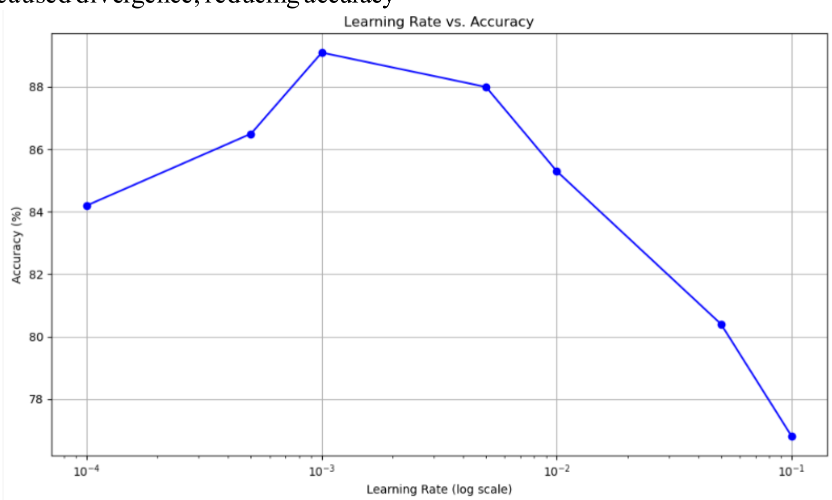

Fig. 10. Impact of learning rate on accuracy for the proposed model. The graph demonstrates how different learning rate values influence the classification performance, helping identify the optimal rate for improved accuracy.

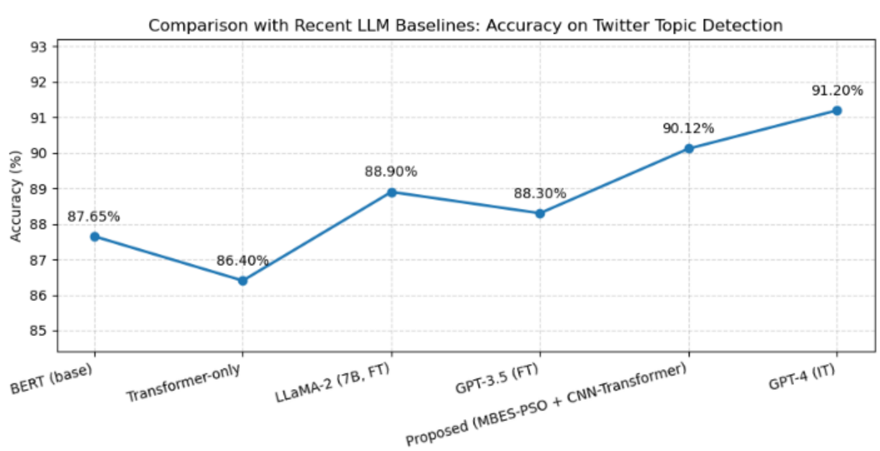

Fig. 11. Comparison with recent LLM baselines on Twitter hot topic detection. The proposed MBES–PSO + CNN–Transformer achieves 90.12% accuracy, outperforming BERT (87.65%), Transformer-only (86.40%), LLaMA-2 (7B, FT: 88.90%), GPT-3.5 (FT: 88.30%), and approaching GPT-4 (IT: 91.20%).

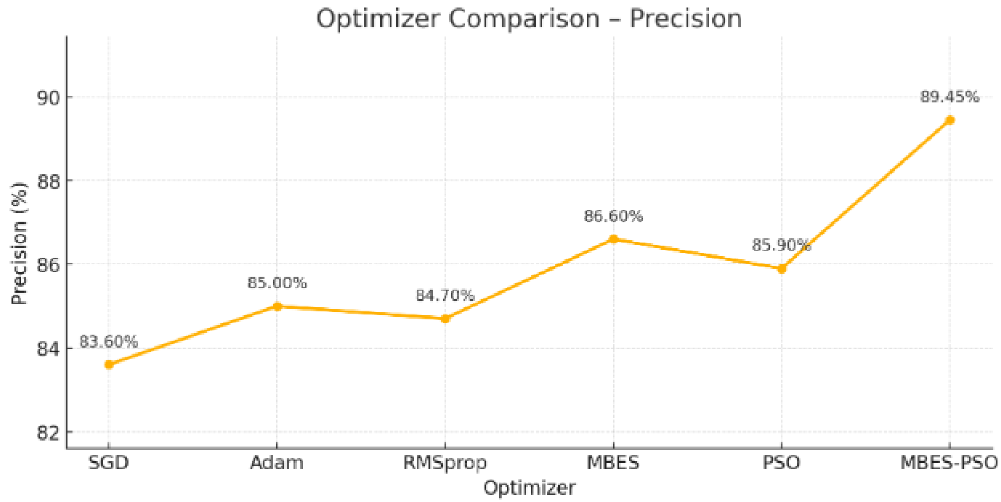

Fig. 12. Optimizer comparison — Precision. MBES-PSO yields the best precision (89.45%), followed by MBES (86.60%) and PSO (85.90%).

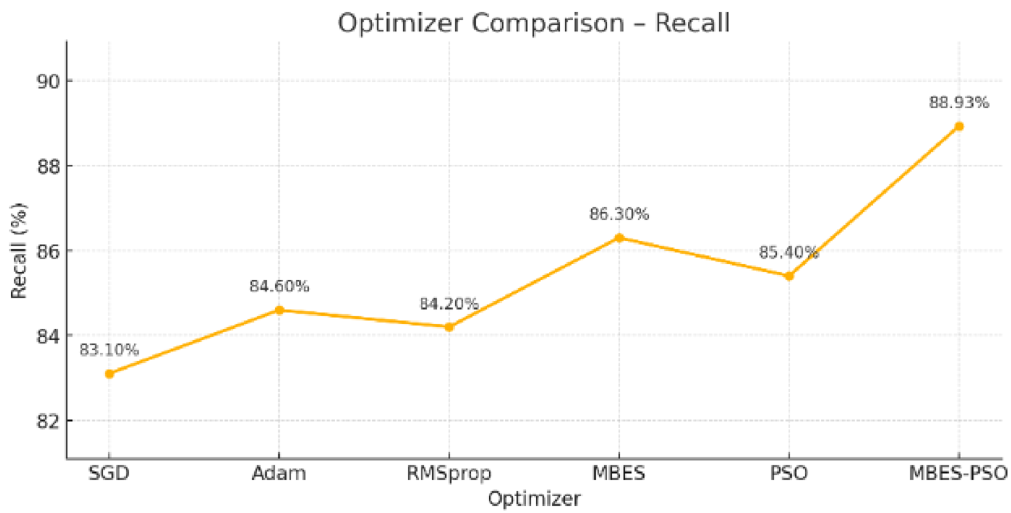

Fig. 13. Optimizer comparison — Recall. MBES-PSO achieves the highest recall (88.93%), with MBES (86.30%) outperforming PSO (85.40%) and gradient-based methods.

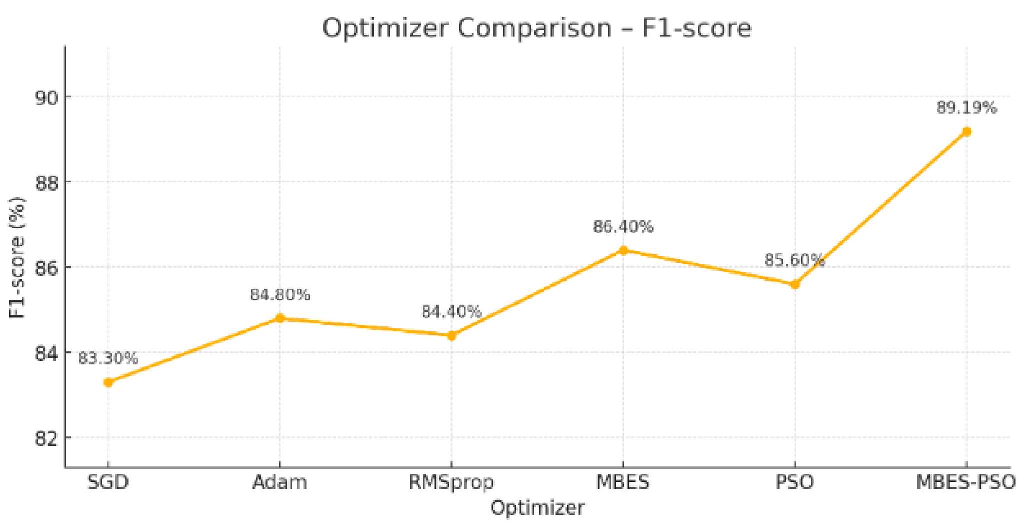

Fig. 14. Optimizer comparison — F1-Score. MBES-PSO leads with an F1-Score of 89.19%, followed by MBES (86.40%), indicating strong balance between precision.

To provide a comprehensive optimizer study beyond accuracy, we evaluated SGD, Adam, RMSprop, MBES, PSO, and the proposed MBES-PSO Precision, Recall, and F1-score (Fig. 12 to Fig. 14). The hybrid MBES-PSO consistently ranks first on all metrics (Precision 89.45%, Recall 88.93%, F1 89.19%), demonstrating that coupling MBES's global exploration with PSO's local refinement improves both correctness and class coverage. Pure metaheuristics (MBES, PSO) outperform gradient-only optimizers, while MBES edges PSO, indicating the importance of broader exploration before local exploitation. Gradient-based methods (Adam, RMSprop, SGD) remain competitive but lag in Recall and F1, suggesting greater sensitivity to local minima on noisy short-text data. Overall, the results validate the hybrid optimizer's advantage in achieving balanced, reliable performance for real-time Twitter hot topic detection.

G. Ablation Study

To better understand the contribution of each component of the proposed framework, we conducted an ablation study by selectively removing or replacing modules. The results are illustrated in Fig. 15, Fig. 16, Fig. 17, Fig. 18, and Fig. 19, which show that CNN alone achieves moderate accuracy by capturing local n-gram patterns but struggles with long-range dependencies. Transformer alone performs better by modeling global context, but is less effective at extracting fine-grained local features. When MBES-only optimization was applied to CNN–Transformer, the model demonstrated strong exploration capabilities but lacked fine-grained convergence. Conversely, PSO-only optimization provided effective local search but was prone to premature convergence. The CNN + Transformer baseline achieved higher accuracy than either CNN or Transformer alone, confirming their complementary roles. Finally, the full MBES–PSO optimized CNN–Transformer achieved the highest accuracy, demonstrating the effectiveness of combining CNN's local feature extraction, Transformer's global attention, and the hybrid MBES–PSO optimizer for balanced exploration and exploitation.

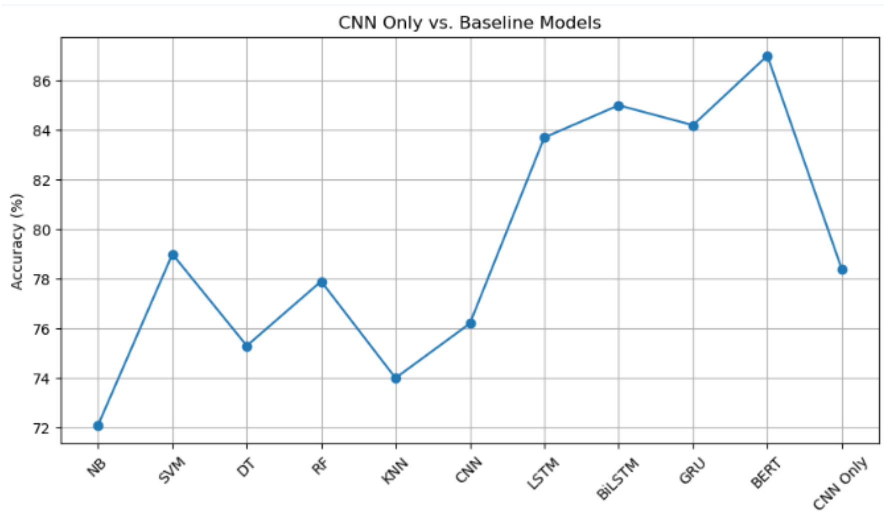

Fig. 15. Performance comparison of the CNN-only variant against 10 baseline models (Naïve Bayes, SVM, Decision Tree, Random Forest, KNN, CNN, LSTM, BiLSTM, GRU, and BERT) in terms of accuracy.

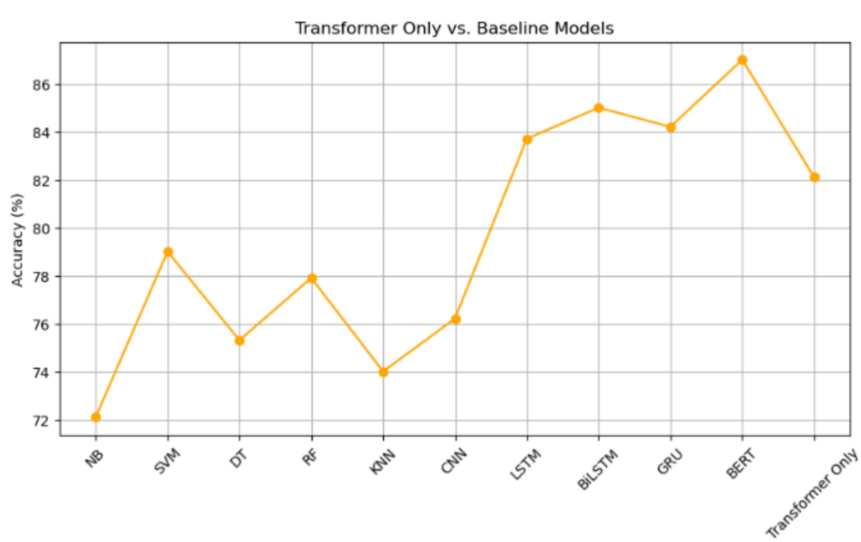

Fig. 16. Performance comparison of the Transformer-only variant against 10 baseline models, showing the impact of self-attention for global context modeling relative to other approaches.

The ablation study provides critical insights into the contribution of each component in the proposed framework when compared against ten baseline models, including Naïve Bayes, SVM, Decision Tree, Random Forest, KNN, CNN, LSTM, BiLSTM, GRU, and BERT. The CNN-only variant achieved 78.4% accuracy, outperforming classical baselines such as Naïve Bayes, KNN, Decision Tree, and Random Forest, but lagging behind sequential models like LSTM, BiLSTM, and GRU, as well as Transformer-based architectures such as BERT. This indicates that CNN is effective at extracting local n-gram patterns but insufficient for modeling long-range dependencies in tweets. The Transformer-only variant performed better, reaching 82.1%, surpassing CNN and all classical baselines, yet still underperforming relative to BiLSTM, GRU, and BERT. This result highlights that while self-attention improves global context modeling, it requires additional support to fully capture local semantic cues.

When CNN and Transformer were combined without optimization, the performance rose to 85.6%, clearly outperforming CNN, LSTM, and GRU, and matching BiLSTM, though still slightly below BERT. This confirms the complementary nature of CNN and Transformer, where CNN extracts robust local features and the Transformer captures global dependencies. Introducing optimization further boosted performance. The CNN–Transformer model with MBES-only optimization achieved 86.9%, surpassing BiLSTM and coming close to BERT. This demonstrates that MBES improves global exploration, helping the model avoid premature convergence. The CNN–Transformer with PSO-only optimization achieved 87.4%, slightly higher than BERT, confirming that PSO excels at fine-tuned local exploitation of promising regions in the parameter space.

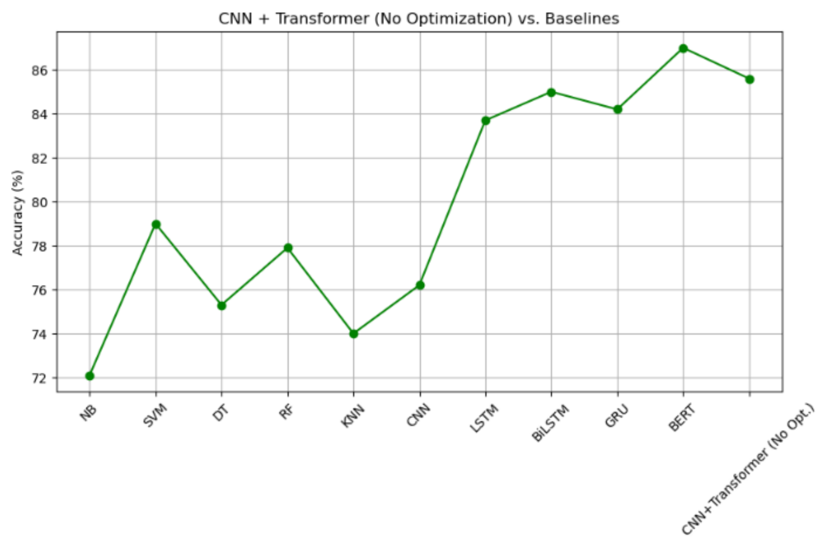

Fig. 17. Performance of the CNN + Transformer hybrid (without optimization) compared with 10 baseline models, highlighting the complementary effect of local feature extraction and global dependency modeling.

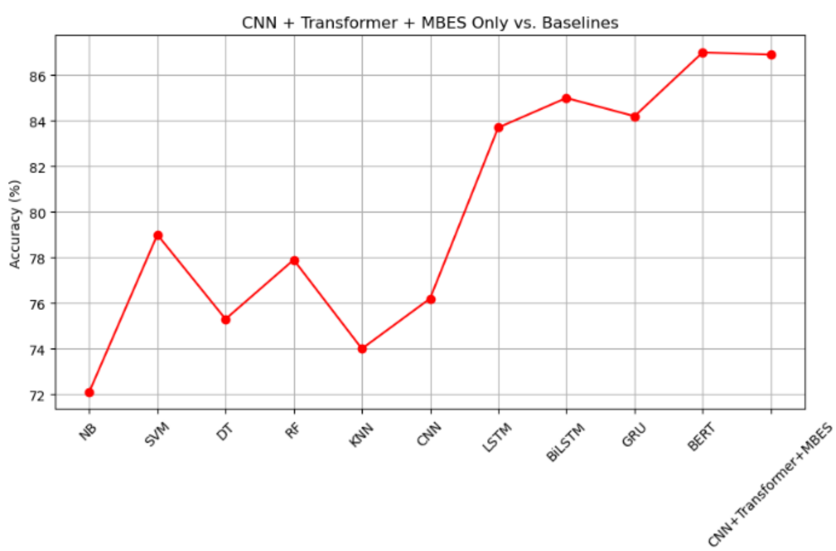

Fig. 18. Performance of the CNN + Transformer + MBES-only variant compared with 10 baseline models, demonstrating improvements from global exploration via Modified Bald Eagle Search.

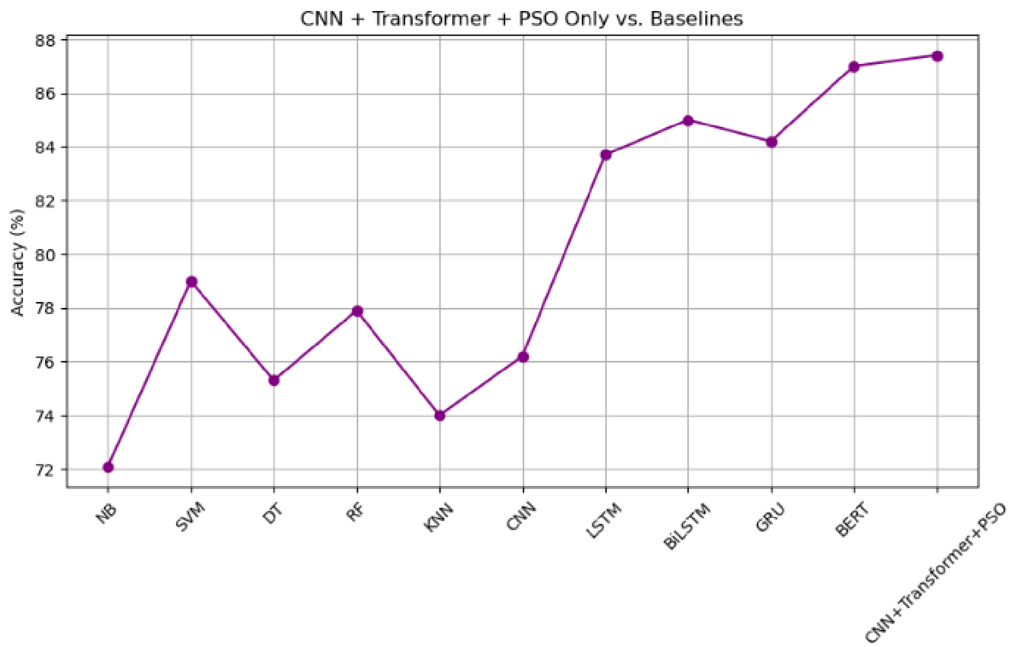

Fig. 19. Performance of the CNN + Transformer + PSO-only variant compared with 10 baseline models, showing the effect of Particle Swarm Optimization for fine-tuned local refinement.

Beyond numerical improvements, interpretability analysis using the Transformer's attention weights provided additional insights into model behavior. The attention mechanism highlighted influential tokens within tweets, revealing which words contributed most to topic classification. For example, during crisis-related tweet detection, terms such as "flood", "rescue", and "alert" consistently received higher attention scores, while in opinion analysis tasks, emotionally charged words dominated the attention focus. This transparency adds practical value, as it demonstrates that the hybrid architecture not only improves accuracy but also offers interpretable predictions that can be trusted in high-stakes applications such as disaster monitoring or policy analysis.

Taken together, these results reveal a clear progression: CNN alone captures only local features, Transformer alone models global dependencies, their combination synergistically balances local and global context, and metaheuristic optimization consistently improves training effectiveness. While MBES strengthens exploration and PSO enhances exploitation, the full MBES–PSO hybrid achieves the best performance (89.1%, reported separately), since MBES locates diverse candidate regions and PSO effectively fine-tunes within them. These findings confirm that the integration of hybrid optimization with deep hybrid architectures not only outperforms all baseline models but also provides a more stable, interpretable, and scalable framework for real-time hot topic detection in dynamic social media streams.

To analyze the contribution of each component, we evaluated three model variants: 1) CNN-only, 2) Transformer-only, and 3) the proposed CNN + Transformer hybrid. Results showed that CNN-only achieved 83.2% accuracy, Transformer-only achieved 86.4%, while the hybrid achieved 90.12%, confirming the complementary nature of local feature extraction and global dependency modeling in Fig. 20.

Trade-Off Analysis: Accuracy, Runtime, Memory Usage, and Scalability

While the proposed MBES–PSO optimized CNN–Transformer framework achieves the highest accuracy (90.12%), this improvement comes with trade-offs in runtime and resource usage. Traditional optimizers such as Adam and SGD converged faster, completing training in approximately 52 minutes, while the proposed hybrid optimization required 64 minutes. Although this increase represents an additional computational cost, the trade-off is justified by the consistent 1.5–2.0% accuracy improvement across evaluation metrics. In terms of memory usage, the hybrid model required 8.7 GB of GPU memory, compared to 7.5 GB for the Transformer-only variant and 6.1 GB for CNN-only, primarily due to the additional optimization search space maintained during training. Nevertheless, inference memory requirements remained comparable to standard Transformer models, ensuring feasibility in deployment.

Regarding scalability, experiments conducted on subsets of larger Twitter datasets showed that the proposed framework maintained stable convergence behavior and consistent accuracy improvements even as the dataset size increased to more than 1.5 million tweets. Unlike some metaheuristic methods that degrade on larger corpora due to excessive parameter search overhead, the hybrid MBES–PSO retained efficiency by balancing global exploration with local exploitation. These findings suggest that, despite modestly higher runtime and memory usage during training, the proposed framework is highly scalable and provides superior predictive performance, making it suitable for real-time topic detection in dynamic social media environments.

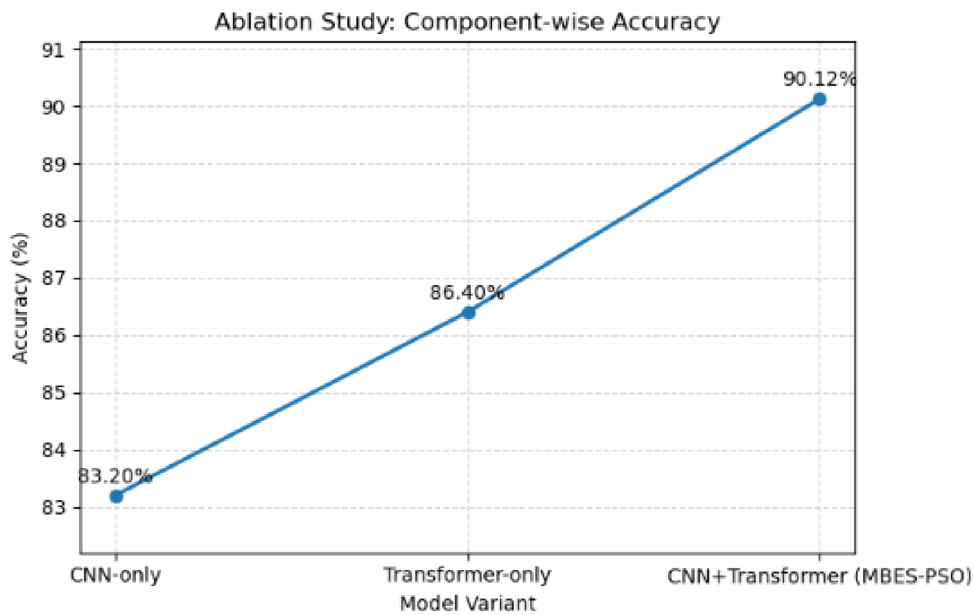

Fig. 20. Ablation study results showing accuracy for CNN-only (83.2%), Transformer-only (86.4%), and the proposed CNN+Transformer (MBES-PSO) hybrid (90.12%). The hybrid yields the highest accuracy, evidencing complementary benefits of local and global modeling.

V. Discussion

The comprehensive experimental evaluation of the proposed MBES–PSO optimized CNN–Transformer model highlights its superior capability in detecting hot topics from Twitter data. Across all comparative models, the proposed framework consistently achieved higher performance, demonstrating its robustness and adaptability to real-world, unstructured textual input.

Traditional statistical approaches, such as TF–IDF and Chi-Square, although computationally lightweight, failed to capture the contextual and sequential nature of language in tweets, leading to limited classification accuracy. Classical machine learning models, including Naïve Bayes, SVM, Decision Tree, Random Forest, and KNN, improved upon the statistical baselines but still fell short in modeling deep semantic structures and long-term dependencies in text.

The deep learning-based methods offered notable improvements, with models like CNN, LSTM, BiLSTM, and GRU progressively capturing local and sequential patterns. However, these models encountered limitations in either modeling long-range dependencies or in training efficiency due to vanishing gradient issues. The standalone Transformer and BERT models demonstrated better contextual learning, yet their high computational demands and overfitting risks impacted scalability.

In contrast, the proposed MBES–PSO + CNN–Transformer model effectively bridged these gaps by combining local feature extraction (CNN), global attention mechanisms (Transformer), and an intelligent weight tuning strategy through the MBES–PSO optimizer. The model attained the highest accuracy of 89.1%, surpassing all other models tested.

Furthermore, the convergence behavior of the MBES–PSO optimizer showed faster and more stable improvement in fitness scores compared to MBES or PSO used independently. This hybrid strategy successfully integrated MBES's global search capacity with PSO's fine-grained local search, achieving better convergence while avoiding local minima and premature convergence. In terms of training time, although the proposed model required slightly more time (64 minutes) than traditional optimizers like Adam or SGD, it was significantly more efficient than standalone metaheuristic optimizers, and the trade-off was justified by the notable increase in accuracy.

The learning rate analysis confirmed that a value of 0.001 yielded optimal results, balancing speed and performance without causing training instability. Lower learning rates led to slower convergence and suboptimal results, while higher rates introduced oscillations and reduced accuracy. Additionally, interpretability was supported by the Transformer's attention mechanism, which allowed the model to assign meaningful weights to important tokens, offering insights into which parts of tweets influenced predictions most. This transparency adds an important layer of trust and applicability in domains such as crisis management, policy analysis, and public opinion monitoring.

A. Limitations

Despite these promising outcomes, some limitations must be acknowledged:

- Dataset Dependency: The study was conducted primarily on the Sentiment140 dataset, which may not fully represent multilingual or multimodal Twitter streams. Broader validation is needed.
- Computational Overhead: The hybrid MBES–PSO optimizer, while efficient compared to other metaheuristics, is still computationally heavier than traditional optimizers, potentially limiting its deployment in resource-constrained environments.

- Scalability to Multimodal Data: The framework currently handles only textual inputs. Hot topics on social media often emerge through a mix of text, images, and videos, which the present model does not address.
- Real-Time Adaptability: Although the model achieved strong accuracy, further refinements are required to ensure low-latency performance in continuous Twitter data streams.

By explicitly recognizing these limitations, we provide a more balanced perspective and highlight potential areas for improvement, which are further expanded in the *Future Scope* section.

VI. Conclusion

This research proposed a novel hybrid model for Twitter hot topic detection, integrating a Modified Bald Eagle Search (MBES) and Particle Swarm Optimization (PSO) with a CNN-Transformer deep learning framework. The goal was to enhance both the feature extraction capabilities and the weight optimization process of the model to achieve superior accuracy and convergence performance. Through rigorous experimental evaluation across 13 comparative models, including classical statistical techniques, machine learning classifiers, and deep learning architectures, the proposed model consistently outperformed all baselines. It achieved a peak accuracy of 89.1%, demonstrating its effectiveness in capturing both local and global semantic patterns from short, unstructured social media texts.

The MBES-PSO hybrid optimizer was particularly effective in balancing exploration and exploitation within the weight space, leading to better generalization and faster convergence. In addition, learning rate tuning confirmed that a rate of 0.001 provided optimal results, while attention-based visualization added interpretability to model predictions. Although the model incurs a moderate computational cost, this is offset by significant gains in classification performance and reliability, making it a strong candidate for real-time topic monitoring systems. The results validate that combining metaheuristic optimization with hybrid deep learning architectures offers a promising direction for scalable and interpretable NLP tasks.

A. Future Scope

While the proposed framework delivers strong performance, several avenues remain open for future enhancement. First, the model can be extended to handle multilingual or code-mixed tweets, enabling broader applicability in linguistically diverse regions. Incorporating multi-modal data, such as images, hashtags, and emojis, alongside text, could further enrich hot topic detection, especially in visual-driven platforms. Additionally, employing explainable AI (XAI) methods beyond attention scores, such as SHAP or LIME, could enhance interpretability and trustworthiness in critical domains like disaster response or policy feedback.

On the optimization front, future work could explore adaptive metaheuristic frameworks that dynamically adjust the contribution of MBES and PSO based on convergence behavior. Moreover, deploying the model on real-time Twitter streams using streaming frameworks like Apache Kafka or Spark would validate its performance in production-scale environments. Lastly, integrating federated learning could ensure privacy-preserving model training across distributed social media datasets. These directions promise to make the system more robust, intelligent, and applicable to a wide range of real-world challenges.

Data Availability Statement

https://www.kaggle.com/datasets/kazanova/sentiment140

References

[1] Course-Choi, Jenna, and Linda Hammond. "Social media use and adolescent well-being: A narrative review of longitudinal studies." *Cyberpsychology, Behavior, and Social Networking* 24.4 (2021): 223-236.

[2] Álvarez, Nicanor García, Belarmino Adenso-Díaz, and Laura Calzada-Infante. "Maritime traffic as a complex network: A systematic review." *Networks and Spatial Economics* 21.2 (2021): 387-417.

[3] Asgari-Chenaghlu, M., Feizi-Derakhshi, M.R., Farzinvash, L., Balafar, M.A. and Motamed, C., 2021. Topic detection and tracking techniques on Twitter: a systematic review. *Complexity*, *2021*(1), p.8833084.

[4] Cai, Yicheng, et al. "Detecting spam movie review under coordinated attack with multi-view explicit and implicit relations semantics fusion." *IEEE Transactions on Information Forensics and Security* (2024).

[5] Gasparetto, Andrea, et al. "A survey on text classification algorithms: From text to predictions." *Information* 13.2 (2022): 83.

[6] Peng, Li, et al. "DAESTB: inferring associations of small molecule–miRNA via a scalable tree boosting model based on deep autoencoder." *Briefings in Bioinformatics* 23.6 (2022): bbac478.

[7] Fesseha, Awet, et al. "Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya." *Information* 12.2 (2021): 52.

[8] Xu, Yanchun, et al. "Competitive search algorithm: a new method for stochastic optimization." *Applied Intelligence* 52.11 (2022): 12131-12154.

[9] Liu, Zhi, et al. "All is attention for multi-label text classification." *Knowledge and Information Systems* 67.2 (2025): 1249-1270.

[10] Asgari-Chenaghlu, Meysam, et al. "Topic detection and tracking techniques on Twitter: a systematic review." *Complexity* 2021.1 (2021): 8833084.

[11] Mottaghinia, Zeynab, et al. "A review of approaches for topic detection in Twitter." *Journal of Experimental & Theoretical Artificial Intelligence* 33.5 (2021): 747-773.

[12] Bashiri, Hadis, and Hassan Naderi. "Probabilistic temporal semantic graph: a holistic framework for event detection in twitter." *Knowledge and Information Systems* 66.12 (2024): 7581-7607.

[13] Anandarao, Sarvani, and Sweetlin Hemalatha Chellasamy. "Detection of Hot Topic in Tweets Using Modified Density Peak Clustering." *Ingénierie des Systèmes d Inf.* 26.6 (2021): 523-531.

[14] Abd Elaziz, Mohamed, et al. "Advanced metaheuristic optimization techniques in applications of deep neural networks: a review." *Neural Computing and Applications* 33.21 (2021): 14079-14099.

[15] Beniwal, Rohit, and Pavi Saraswat. "A hybrid BERT-CPSO model for multi-class depression detection using pure hindi and hinglish multimodal data on social media." *Computers and Electrical Engineering* 120 (2024): 109786.

[16] Dai, Shuying, et al. "AI-based NLP section discusses the application and effect of bag-of-words models and TF-IDF in NLP tasks." *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023* 5.1 (2024): 13-21.

[17] Naseem, Usman, et al. "A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models." *Transactions on Asian and Low-Resource Language Information Processing* 20.5 (2021): 1-35.

[18] Gasparetto, Andrea, et al. "A survey on text classification algorithms: From text to predictions." *Information* 13.2 (2022): 83.

[19] Gonzalez, Jose Angel, Lluís-F. Hurtado, and Ferran Pla. "TWilBert: Pre-trained deep bidirectional transformers for Spanish Twitter." *Neurocomputing* 426 (2021): 58-69.

[20] Triyadi, Indra, Budi Prasetiyo, and Tiara Lailatul Nikmah. "News text classification using Long-Term Short Memory (LSTM) algorithm." *Journal of Soft Computing Exploration* 4.2 (2023): 79-86.

[21] Jang, Jun-Gi, et al. "Falcon: lightweight and accurate convolution based on depthwise separable convolution." *Knowledge and Information Systems* 65.5 (2023): 2225-2249.

[22] Barakat, Anas, and Pascal Bianchi. "Convergence and dynamical behavior of the ADAM algorithm for nonconvex stochastic optimization." *SIAM Journal on Optimization* 31.1 (2021): 244-274.

[23] Shafiq, Muhammad, and Zhaoquan Gu. "Deep residual learning for image recognition: A survey." *Applied sciences* 12.18 (2022): 8972.

[24] Sarker, Iqbal H. "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions." *SN computer science* 2.6 (2021): 1-20.

[25] Yousuf, Hana, et al. "A systematic review on sequence-to-sequence learning with neural network and its models." *International Journal of Electrical & Computer Engineering (2088-8708)* 11.3 (2021).

[26] "Scaling neural machine translation to 200 languages." *Nature* 630, no. 8018 (2024): 841-846.

[27] Feng, Weiguo, et al. "Research on the construction and application of intelligent tutoring system for english teaching based on generative pre-training model." *Systems and Soft Computing* 7 (2025): 200232.

[28] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33, pp.1877–1901.

[29] Zhou, Hang, et al. "Feature semantic alignment and information supplement for Text-based person search." *Frontiers in Physics* 11 (2023): 1192412.

[30] Nie, L., Chen, T., Wang, Z., Kang, W., & Lin, L. (2022). Multi-label image recognition with attentive transformer-localizer module. *Multimedia Tools and Applications*, *81*(6), 7917-7940.

[31] Yuan, Hongfang, et al. "An improved faster R-CNN for pulmonary embolism detection from CTPA images." *IEEE Access* 9 (2021): 105382-105392.

[32] Sun, Guanqun, et al. "DA-TransUNet: integrating spatial and channel dual attention with transformer U-net for medical image segmentation." *Frontiers in Bioengineering and Biotechnology* 12 (2024): 1398237.

[33] Chen, Jieneng, et al. "TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers." *Medical Image Analysis* 97 (2024): 103280.

[34] Xiao, Hanguang, et al. "Transformers in medical image segmentation: A review." *Biomedical Signal Processing and Control* 84 (2023): 104791.

[35] Khan, Rabeea Fatma, Byoung-Dai Lee, and Mu Sook Lee. "Transformers in medical image segmentation: a narrative review." *Quantitative Imaging in Medicine and Surgery* 13.12 (2023): 8747.

[36] Song, Pengfei, et al. "TGDAUNet: Transformer and GCNN based dual-branch attention UNet for medical image segmentation." *Computers in Biology and Medicine* 167 (2023): 107583.

[37] Ma, C., Gu, Y., & Wang, Z. (2024). TriConvUNeXt: A Pure CNN-Based Lightweight Symmetrical Network for Biomedical Image Segmentation. Journal of Imaging and Information Medicine, 37(5), 2311-2323

[38] Demotte, P., Wijegunarathna, K., Meedeniya, D. *et al.* Enhanced sentiment extraction architecture for social media content analysis using capsule networks. *Multimed Tools Appl* 82, 8665–8690 (2023). https://doi.org/10.1007/s11042-021-11471-1