ISSN 2723-6471

Enhancing Support Vector Machines Accuracy Through Firefly Algorithm-Driven Feature Optimization for Forest-Fire Sentiment Classification

Wafa Salma Sentanu[1], Dinar Ajeng Kristiyanti[2,*]

[1,2]*Information System Study Program, Universitas Multimedia Nusantara, Tangerang, 15810, Indonesia*

(Received: February 1, 2026; Revised: March 25, 2026; Accepted: June 20, 2026; Available online: July 20, 2026)

Abstract

Forest fires are a growing global environmental issue that significantly impact ecosystems, human health, and exacerbate climate change. Data from 2023 indicate that approximately 11.91 million hectares of forest were lost due to fires, highlighting the urgent need for technological approaches in addressing this issue. One relevant approach is public sentiment analysis based on the social media platform X, which enables the rapid and real-time capture of public perception. In text-based sentiment analysis, key challenges include high-dimensional feature space and class imbalance, which can degrade the performance of machine learning algorithms. Therefore, this study applies feature selection methods based on swarm intelligence, namely Particle Swarm Optimization (PSO) and Firefly Algorithm (FA), to enhance classification efficiency and accuracy. The models evaluated include Support Vector Machine (SVM), Naïve Bayes (NB), and K-Nearest Neighbors (KNN), using the Knowledge Discovery in Databases (KDD) approach, which involves selection, preprocessing, transformation with Term Frequency-Inverse Document Frequency (TF-IDF) and Synthetic Minority Oversampling Technique (SMOTE), and data splitting with an 80:20 ratio. Model performance was evaluated using four metrics: accuracy, precision, recall, and F1-score. The models achieved strong performance on the balanced training data, indicating effective learning after feature selection, with the fastest execution time recorded by FA optimization at 0.0071 seconds. To ensure a fair assessment of generalization, the main conclusions of this study are based on the testing results. On the testing data, SVM with FA achieved the highest accuracy of 96.36% with an execution time of 0.0064 seconds. Overall, swarm intelligence-based feature selection (PSO and FA) enhances the efficiency of conventional classifiers by reducing high-dimensional feature representations and execution time while maintaining strong predictive performance for forest-fire sentiment classification.

Keywords: Firefly Algorithm, Forest Fires, Machine Learning, Particle Swarm Optimization, Sentiment Analysis, Support Vector Machines

1. Introduction

Forest fires are one of the most severe environmental problems, affecting ecosystems, public health, and the global economy. In addition to causing tree cover loss, forest fires result in significant economic losses, peaking in 2018 at over 22 billion USD [1]. According to the third session of the Intergovernmental Panel on Climate Change (IPCC) Sixth Assessment Report (AR6), the area of forest lost due to fire continues to increase, reaching 11.91 million hectares in 2023 [3]. These disasters highlight the crucial role of technology in detecting, predicting, and responding quickly to forest fires to protect people and the environment. Forest fires not only affect ecological and economic aspects but also receive widespread attention on social media.

Social media has become a crucial platform for monitoring public sentiment on environmental disasters, especially forest and land fires [4]. X platform (formerly Twitter) plays a key role due to its real-time updates and wide global reach [5]. These online discussions provide rich data for sentiment analysis, allowing researchers and policymakers to assess public concerns and develop more effective mitigation and awareness strategies. One effective way to achieve this is by conducting sentiment analysis of public opinion related to forest and land fire incidents. In sentiment analysis, three common approaches are widely used: lexicon-based, machine learning-based, and hybrid [2]. Among these, the machine learning-based approach has gained popularity for its ability to automatically learn patterns from data and classify text into sentiment categories typically positive, negative, or neutral using supervised algorithms such as SVM, Naïve Bayes, and KNN [6].

***Corresponding author: Dinar Ajeng Kristiyanti (dinar.kristiyanti@umn.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i3.1144

Although these conventional classifiers have shown promising performance in large-scale social media datasets, high-dimensional TF-IDF representations can increase computational burden and reduce efficiency. Moreover, noisy and redundant textual features may lead to overfitting and reduce model robustness in real-world settings [7]. To address these issues, hybrid methods combining machine learning and feature selection have been introduced. Feature selection helps identify relevant features while removing redundant ones, improving both effectiveness and efficiency [8]. Recent studies have demonstrated the effectiveness of swarm intelligence-based methods such as PSO and FA for optimizing feature selection in sentiment classification [9]. Additionally, to handle class imbalance commonly found in social media data, the SMOTE is applied only to the training set after the train-test split, ensuring balanced learning without introducing data leakage and preserving a fair evaluation on the untouched test set [10]. Following this approach, the present study integrates PSO and FA as feature selection techniques with SMOTE preprocessing to improve both classification efficiency and performance on global forest fire discussions from the X platform.

These findings affirm the potential of swarm intelligence-based optimization, particularly PSO and FA, in enhancing the efficiency and reliability of conventional classifiers for sentiment analysis. By combining SMOTE with swarm-based feature selection, the present study provides a novel contribution to the domain of global forest fire sentiment classification on social media. This study explores the application of PSO and FA as feature selection techniques to enhance sentiment classification related to forest and land fire incidents on the X platform.

In addition to improving data balance, researchers have increasingly explored the use of optimization techniques based on swarm intelligence, particularly PSO and the FA, to enhance both accuracy and computational efficiency. For example, a study proposed a Chaotic Firefly Algorithm (CFA) for feature selection in Arabic text categorization and reported that CFA combined with SVM consistently achieved the highest accuracy, nearly 96% across various datasets [11]. Similarly, another study demonstrated that integrating FA with SVM significantly improved classification precision from 97.00-98.00% to 99.40% in improving Arabic text classification accuracy [12]. PSO has also shown strong performance; it was applied to both SVM and Naïve Bayes, resulting in 91.30% accuracy for SVM and 93.10% for Naïve Bayes, with the SVM model achieving an AUC of 0.970 [13]. Additionally, a hybrid voting model combining Random Forest, SVM, and KNN with PSO and Genetic Algorithm (GA) achieved accuracy up to 96.89% across three benchmark datasets [14].

These findings affirm the superiority of swarm intelligence-based optimization, particularly PSO and FA, in enhancing both the performance and computational efficiency of conventional classifiers for sentiment analysis. The implementation of SMOTE combined with swarm-based feature selection, the present study provides a novel contribution to the domain of forest fire sentiment classification on global social media. This study explores the application of metaheuristic optimization algorithms, specifically PSO and FA, as feature selection techniques to enhance sentiment classification related to forest and land fire on the X platform. The key contributions of this research are as follows: 1) implementing FA and PSO as feature selection optimizers in sentiment analysis tasks, 2) utilizing a large-scale dataset comprising 10,500 English-language tweets related to forest and land fires collected from the X platform, 3) conducting a comparative analysis between FA and a competing algorithm, PSO, to evaluate classification performance, and 4) applying the SMOTE technique only to the training set after data splitting to address class imbalance and improve model generalization without data leakage.

The structure of this paper is arranged as follows: Section 1 presents the background of the study, describing the research context and objectives. Section 2 delivers a literature review, highlighting related studies and their relevance to this work. Section 3 explains the research methodology, including data sources, preprocessing, and modeling steps. Section 4 provides the main analysis and findings, focusing on the outcomes of sentiment classification. Section 5 discusses the results in depth, interpreting their implications and comparing them with prior studies. Finally, Section 6 concludes the research by summarizing key insights and offering suggestions for future work.

2. Literature Review

Forest fires are among the most visible consequences of climate change, driven by rising global temperatures and prolonged drought that create drier conditions and accelerate CO_2 accumulation in the atmosphere [15]. These disasters not only endanger ecosystems but also result in severe social and economic impacts, including health issues and high firefighting costs. In recent years, researchers have increasingly utilized social media platforms, especially X platform,

to analyze public sentiment and awareness toward environmental issues in real time [5]. Sentiment analysis has therefore become an essential tool for understanding public perception and guiding more effective environmental policies. Sentiment analysis classifies opinions expressed in text into positive, negative, or neutral categories [16]. In this study, the VADER lexicon is used for labeling, as it performs well in handling short and informal text typical of social media. For classification, three machine learning algorithms are implemented: Naïve Bayes for probabilistic modeling, SVM for high-dimensional data classification, and KNN for distance-based categorization. These algorithms have demonstrated robust performance in prior sentiment analysis research.

To further enhance model performance, feature engineering and optimization techniques are employed. The TF-IDF method transforms textual data into numerical form suitable for machine learning, while SMOTE addresses class imbalance commonly found in social media datasets [16]. Additionally, this study integrates PSO and FA as feature selection methods. PSO, inspired by bird flock movements, and FA, based on firefly attraction behavior, both aims to identify the most relevant features while reducing dimensionality and computational cost [17]. The integration of these optimization techniques is expected to improve accuracy, efficiency, and overall reliability of sentiment classification models.

3. Methodology

This study adopts the Knowledge Discovery in Databases (KDD) framework, which consists of a structured process for extracting useful knowledge from large datasets [18]. The KDD process in this research involves 5 key stages: data selection, data pre-processing, data transformation, data mining, and evaluation. Each stage plays a distinct role in preparing the data and extracting insights through sentiment classification. The overall research flow following the KDD methodology is illustrated in figure 1.

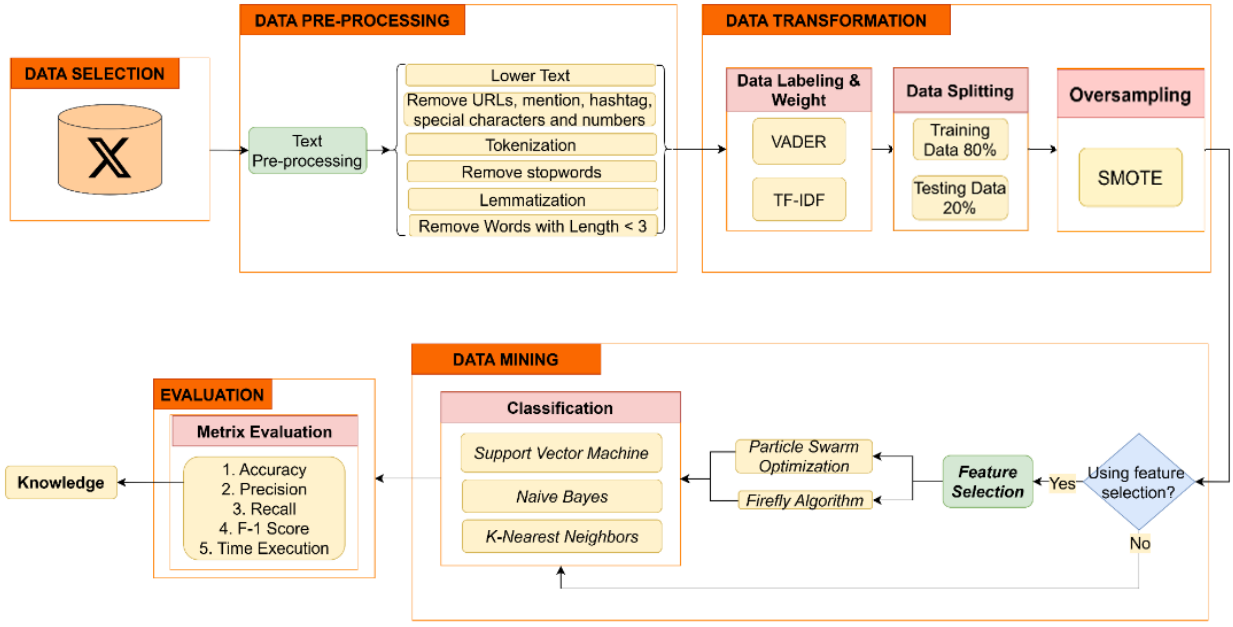

Figure 1. Stages of Research

Figure 1 illustrates the stages of the KDD process applied in this study, which aims to analyze public sentiment related to forest fires using X platform data. The KDD framework serves as a systematic approach to extract meaningful insights from raw data through a sequence of essential steps.

3.1. Data Selection

The initial step in this study involves the selection of data related to public opinions on forest fires, specifically collected from X platform. The X platform API is utilized to extract both real-time and historical tweets using keyword-based queries such as “forest fire” and “wildfire”. This approach enables the retrieval of dynamic and large-scale public discourse, which is essential for understanding real-time sentiment trends during fire-related events. X platform is

chosen due to its high data volume, public accessibility, and frequent use during disaster events, making it a reliable source for capturing community responses and opinions [19]. To ensure the relevance of the dataset, filtering based on keywords helps eliminate unrelated content and focuses only on tweets directly associated with forest fire incidents.

To maintain consistency and relevance in the analysis, the data gathered is restricted to tweets written in English. This language selection is intended to target a broader, global audience and to facilitate the application of existing natural language processing tools, which are often optimized for English-language content. Furthermore, the timeframe for data collection spans one full year, from January 1, 2024, to January 31, 2025. This period was chosen to obtain the most recent and up-to-date public commentary, capturing discourse surrounding forest fires across different seasons and regions.

Figure 2 illustrates the data collection workflow using secondary data, which refers to information gathered indirectly from the source. The data is obtained through a scraping process on platform X using Python, executed within Google Colaboratory. The collected tweets are stored in a `.csv` file named `Data_X.csv`. This scraping process employs the `tweet-harvest` library, which integrates with Node.js to enable continuous data extraction despite platform limitations. Table 1 presents the list of variables included in the dataset.

Figure 2. Data Collection Technique

Table 1 displays the key variables included in the dataset. These variables cover tweet metadata such as the timestamp, content, language, and engagement metrics including quotes, replies, retweets, and likes. Additionally, unique identifiers for both the tweet and the user are included to support data tracking and analysis.

Table 1. Variable Dataset

Attribute Name	Data Type	Description
created_at	Date	Timestamp when the tweet was posted.
id_str	String	Unique identifier for the tweet.
full_text	String	Complete content of the tweet.
quote_count	Numeric	Number of times the tweet was quoted.
reply_count	Numeric	Total replies received by the tweet.
retweet_count	Numeric	Number of times the tweet was retweeted.
favorite_count	Numeric	Total likes received by the tweet.
lang	String	Language used in the tweet.
user_id_str	String	Unique identifier of the tweet's author.

3.2. Data Pre-Processing

Following data acquisition, the tweets undergo a comprehensive pre-processing stage to clean and normalize textual content. The applied pre-processing steps aim to reduce noise, standardize word forms, and improve the quality of features used in downstream models. The preprocessing stage was conducted to improve the quality and consistency of the textual data before feature extraction and classification. First, all characters were converted into lowercase to ensure uniform word representation, so that words such as “Fire” and “fire” were treated as the same term [20]. The text was then cleaned by removing irrelevant elements, including URLs, emojis, hashtags, mentions, punctuation marks, and numbers [20]. After that, tokenization was applied to split each tweet into individual tokens, usually in the form of words [20]. For example, the sentence “Forest fire is dangerous” was transformed into the token’s “forest”,

"fire", "is", and "dangerous". Stopword removal was subsequently performed to eliminate frequently occurring words, such as "the", "is", "at", and "in", which generally do not carry substantial semantic value for sentiment classification [20]. Lemmatization was then applied to reduce each word to its base or dictionary form; for instance, "running", "ran", and "runs" were reduced to "run" [20].

In addition, words with fewer than three characters were removed because very short tokens often contribute little semantic meaning in sentiment analysis [20]. A final stopword re-filtering process was also conducted after lemmatization and short-token removal to ensure that remaining functionally irrelevant tokens were eliminated, thereby reducing residual noise and improving the quality of the feature set. Finally, the cleaned tokens were joined back into text sequences to prepare the data for feature extraction methods such as TF-IDF [21]. This rigorous cleaning process ensures that the dataset is both semantically rich and computationally efficient for further transformation and modeling.

3.3. Data Transformation

The data transformation stage integrates three sequential procedures annotation, text vectorization, and oversampling to prepare the tweets for machine learning modelling [22]. Sentiment annotation was conducted using VADER, a lexicon-based sentiment analysis tool designed for short and informal social media texts. In this phase, each tweet was assigned a sentiment label based on the compound polarity score generated by VADER, which was then mapped into positive, negative, and neutral classes [23]. These sentiment labels are used as target variables for subsequent machine learning models. VADER was chosen for its practicality in large-scale automatic labeling; however, as a general-domain lexicon, it may not fully capture sarcasm, implicit sentiment, cultural nuance, or disaster-specific expressions. Thus, the resulting labels may contain weak-label noise, which represents a common trade-off in large-scale sentiment studies.

Text Vectorizations with TF-IDF was conducted to transform the preprocessed textual data into a numerical representation suitable for machine learning algorithms. In this study, the TF-IDF method was employed to assign weights to terms based on their frequency in a document and their importance across the entire corpus, thereby enabling the model to emphasize more informative and contextually significant terms [24]. Equation (1) presents the formula for calculating the augmented Term Frequency (TF) [24]:

$$tf'(t,d) = 0.5 + 0.5\ x\ \frac{tf(t,d)}{max_{t \in d}\ tf(t',d)} \quad (1)$$

t denotes a term, d denotes a document (tweet), $tf(t,d)$ is the frequency of term t in document d, and $max_{t' \in d}\ tf(t',d)$ is the maximum term frequency of any term t' within document d. This normalization reduces bias toward overly frequent terms and variations in document length. In addition, the formula for calculating IDF (Inverse Document Frequency) is presented in Equation (2) [25]:

$$idf_{(t)} = \log(\frac{\mathrm{D}+1}{df\ (t)+1}) + 1 \quad (2)$$

In Equation (2), the IDF quantifies the importance of a term *t* across a set of documents. Where D is the total number of documents in the corpus and $df(t)$ is the number of documents containing term t. The logarithmic function serves to diminish the weight of terms that appear in many documents, thereby highlighting terms that are more distinctive. This approach is widely used in the TF-IDF scheme to improve the accuracy of text representation in information retrieval and text mining applications.

Data Splitting was performed by dividing the transformed dataset into training and testing subsets to evaluate the model's generalization capability. Following standard practice, this study adopted an 80:20 ratio, where 80% of the data was used to train the model, while the remaining 20% was reserved for testing its predictive accuracy on unseen data [26]. This split ensures a fair evaluation of the model's performance and prevents overfitting.

Oversampling with SMOTE was applied to address class imbalance in the sentiment classification dataset, where one sentiment category may be more dominant than others. To mitigate this issue, the SMOTE algorithm was employed to

ISSN 2723-6471

generate synthetic samples for the minority class, thereby creating a more balanced dataset and improving the model's ability to learn from underrepresented sentiment categories [27]. The dataset was first split into training and testing sets using an 80:20 ratio. To address class imbalance, SMOTE was applied only to the training set. The resulting balanced training data were then transformed using TF-IDF and further optimized using PSO and FA as feature selection techniques. Finally, model performance was evaluated on the original untouched test set using accuracy, precision, recall, and F1-score. This method creates artificial samples of the minority class by interpolating between existing instances, effectively balancing the class distribution and mitigating classifier bias toward the majority class. This improves model robustness and predictive performance. The following equation formula for the SMOTE technique is shown in Equation (3) [28].

$$V_{new} = V_i + (M_i - V_i) \times P \tag{3}$$

In Equation (3), a new synthetic sample is generated along the line segment between a minority class instance and one of its nearest neighbors. The term introduces a controlled variation, where P is a random value between 0 and 1 that determines how close the synthetic point is to either or This interpolation strategy ensures that the newly generated instance remains within the minority class region, thereby improving class balance without introducing noise. This method forms the basis of the SMOTE, which is commonly used to handle imbalanced datasets by increasing the number of minority class samples.

3.4. Feature Selection Techniques

To further enhance the efficiency and accuracy of the classification model, this study incorporates feature selection techniques based on swarm intelligence algorithms, which can identify the most relevant and informative features from high-dimensional datasets, thereby reducing computational complexity and improving predictive performance [29]. Specifically, this study utilizes two swarm intelligence techniques PSO and the FA due to their demonstrated ability to optimize feature subsets and boost classification outcomes.

PSO is a population-based metaheuristic inspired by the collective behavior of animals such as bird flocks or fish schools during foraging. Each particle represents a candidate solution and adjusts its position in the search space based on its own experience and that of the swarm. Movement is influenced by the particle's personal best and the global best positions, allowing the swarm to balance exploration and exploitation in the search for optimal solutions.

PSO is an optimization technique inspired by the behavior of animal swarms, such as birds or fish, during food searches. Each particle represents a potential solution moving through an N-dimensional search space in this method. Particles adjust their position based on their own best experience (Pbest) and the best-known position in the group (Gbest). By updating their velocity using this information, particles collaboratively explore and exploit the search space. The process continues until a stopping criterion is met, making PSO an effective approach for finding optimal or near-optimal solutions. The update rules for PSO are defined by two mathematical formulations that iteratively adjust the velocity and position of each particle [30].

$$v_i^{t+1} = w.v_i^t + c_1 r_1(\vec{pi} - \vec{xi}) + c_2 r_2(\vec{pg} - \vec{xi})$$
$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{4}$$

In Equation (4), each particle updates its velocity vector v_i^{t+1} based on three components: the inertia term $w.v_i^t$, the cognitive component $c_1 r_1(\vec{p}i - \vec{x}i)$, and the social component $c_2 r_2(\vec{p}g - \vec{x}i)$. Here, w is the inertia weight controlling the influence of the previous velocity, c_1 and c_2 are acceleration coefficients, and r_1, r_2 are random numbers uniformly distributed in [0, 1], which introduce stochastic behavior to maintain diversity in the search space. After updating the velocity, the new position x_i^{t+1} is computed by adding the updated velocity to the current position. This mechanism enables each particle to explore the search space by balancing individual experience (personal best $\vec{p}i$ and collective knowledge (global best $(\vec{p}g)$). The detailed pseudocode of the PSO algorithm is presented in table 2.

Table 2. Particle Swarm Optimization Pseudocode [31]

Step	Description
1	Initialize a swarm of particles with random positions and velocities in a D-dimensional space.
2	Start the main loop

ISSN 2723-6471

3	For each particle, calculate its fitness based on the objective function
4	Compare the current fitness of each particle with its previously recorded best (pbesti). If the current result is better: • Update pbesti with the new fitness value. • Update the particle's best-known position (pi) to its current position (xi).
5	Determine the particle with the highest fitness in the neighborhood and set its index as g (global best).
6	Adjust the velocity and position of each particle using the standard PSO update equations.
7	Repeat the loop until a stopping condition is met, such as reaching a fitness threshold or maximum number of iterations.

The FA is a nature-inspired optimization method that mimics the flashing behavior of fireflies. It is based on the idea that fireflies are attracted to brighter individuals, where brightness is associated with the quality or fitness of a solution [32]. Fireflies adjust their positions depending on their relative brightness, which decreases with increasing distance. The algorithm promotes exploration and exploitation by combining deterministic attraction with random movement. This behavior enables the algorithm to effectively search for optimal or near-optimal solutions in complex, high-dimensional spaces. The update rules for FA are [33]:

r_{ij}^2 is the euclidean distance between fireflies i and j, d is the dimensionality of the solution vector, β_0 is attractiveness at $r = 0$, γ is the absorption coefficient, α is the randomization parameter, and ϵ is a random vector. In Equation (5), the new position x_i^{t+1} of firefly i is updated based on its current position x_i^t, the attractiveness toward another brighter firefly j, and a randomization component. The term $\beta_0 e^{-\gamma r_{ij}^2}(x_j^t + x_i^t)$ represents the attractiveness function, where β_0 is the attractiveness at distance zero, γ is the light absorption coefficient, and r_{ij} is the Euclidean distance between firefly i and firefly j. This component models the movement of firefly i toward firefly j based on brightness and distance. The term $\alpha\epsilon$ introduces randomness to enhance exploration, where α is the randomization parameter and ϵ is a vector of random numbers drawn from a uniform or Gaussian distribution. Together, these components allow the fireflies to converge toward brighter and potentially better solutions while maintaining diversity in the population. The detailed pseudocode of the Firefly Algorithm is presented in table 3.

Table 3. Firefly Algorithm Pseudocode [34]

Step	Description
1	Initialize a population of fireflies with random positions in a D-dimensional search space.
2	Set algorithm parameters: light absorption coefficient (γ), attractiveness coefficient (β_0), and randomization parameter (α).
3	Start the main loop:
4	Evaluate the fitness of each firefly based on the objective function.
5	For every pair of fireflies i and j: • If firefly j is brighter than i move firefly i toward j using the movement formula.
6	Apply random movement to enhance diversity in the swarm
7	Update the brightness of each firefly based on the new fitness.
8	Repeat the process until a stopping condition is satisfied (e.g., maximum iterations or desired fitness achieved).

Both PSO and FA are used to reduce dimensionality while preserving classification performance, thereby enhancing the speed and interpretability of the sentiment analysis models. By selecting the most relevant features, these optimization algorithms help eliminate noise and redundant data that could hinder model accuracy. As a result, the models not only perform faster but also become easier to analyze.

After selecting the most relevant features, the study employs three widely used supervised learning algorithms to perform sentiment classification. SVM is a supervised learning method commonly used for classification tasks in machine learning. It can handle both binary and multiclass classification problems and is supported by strong statistical theory. Due to its effectiveness in dealing with large datasets, SVM is considered a powerful yet complex algorithm. It identifies the optimal hyperplane by maximizing the margin between classes. This hyperplane acts as a boundary, with the widest margin known as the Maximum Margin Hyperplane. Two parallel lines on either side, called the Positive and Negative Hyperplanes, touch the nearest data points from each class, known as Support Vectors. These vectors determine the optimal position of the hyperplane. The algorithm begins by analyzing the training data to

separate the classes and identify outliersdata points that fall outside the decision boundary. Equation (6) represents the mathematical formula for the hyperplane in the SVM algorithm [31]:

$$w \cdot x + b = 0 \tag{6}$$

In Equation (6), the SVM defines a decision boundary using the hyperplane formula $w \cdot x + b = 0$. In this equation, w denotes the weight vector that determines the orientation of the hyperplane, while x represents the input feature vector. The dot product operation $w \cdot x$ calculates the projection of the input onto the weight vector. The term b is the bias, which shifts the hyperplane away from the origin. The equation being equal to zero indicates that it defines the hyperplane that separates data points from two different classes. This mathematical formulation allows SVM to construct an optimal separating hyperplane by maximizing the margin between the support vectors of each class.

Naïve Bayes is a relatively simple yet powerful classification algorithm that estimates class probabilities by analyzing the frequency and distribution of feature values within a dataset. It is grounded in Bayes' Theorem and operates under the strong assumption that all features are conditionally independent given the class label [35], meaning that the joint probability of a set of features can be computed as the product of the individual probabilities of each feature. The Naive Bayes algorithm divides the feature space into classification regions based on the posterior probability of each class. It estimates the likelihood that a data point belongs to a particular class by considering the distribution of features within that class. The formula used in this method is presented in Equation (7) [36].

$$P(X) \frac{P(H)\ x\ P(H)}{P(x)} \tag{7}$$

In Equation (7), the Naïve Bayes formula is expressed as $P(H|X) \frac{P(X|H)\ x\ P(H)}{P(X)}$, where $P(H|X)$ is the posterior probability of class H given data X. $P(X|H)$ represents the likelihood, $P(H)$ is the prior probability of the class, and $P(X)$ is the evidence or the overall probability of the input data. This equation enables classification by selecting the class with the highest posterior probability, assuming feature independence.

The KNN algorithm classifies data by identifying the K nearest neighbors to determine the class of a new data point [37]. It operates in a high-dimensional space and does not rely on fixed parameters, making it flexible in classification tasks. The working principle of the KNN algorithm, where a test point (Xu) is classified based on its proximity to the K nearest training data points. The algorithm calculates the distance between the test point and all training data, then selects the K closest neighbors. The class with the highest frequency among these neighbors becomes the predicted class for Xu. In the example, Xu has neighbors from three different classes (W1, W2, and W3) and is assigned to the majority class. One commonly used distance metric is Euclidean distance, which measures the straight-line distance between two points in a two-dimensional space. This is calculated using the Pythagorean theorem, as shown in Equation (8) [38].

$$d(x, y) \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{8}$$

In Equation (8), the Euclidean distance $d(x, y)$ is computed between two vectors x and y by taking the square root of the sum of squared differences between their corresponding components. Here, x_i and y_i represent the i-th feature values of vectors x and y, respectively, while mmm denotes the total number of dimensions or features. This distance metric is commonly used in the KNN algorithm to measure similarity between data points in a multidimensional space.

The final phase of the classification process involves evaluating the model's performance using standard metrics derived from the confusion matrix [39]. These metrics Accuracy, Precision, Recall, and F1-Score in Equation (9-12) are essential for understanding various aspects of a model's predictive capabilities. Accuracy is the most used metric to evaluate classification performance due to its straightforward interpretation. It represents the ratio of correctly predicted instances to the total number of predictions made. Although widely adopted, accuracy alone may be insufficient when dealing with imbalanced datasets, as it may provide a misleading impression of model performance [17].

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{9}$$

Precision is a critical metric that focuses on the quality of positive predictions. It quantifies how many of the predicted positive instances are truly positive. This measure is particularly important in contexts where false positives carry significant consequences, such as spam detection or medical diagnosis [17].

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

Recall measures the model's ability to detect all actual positive instances within a dataset. It is especially important in scenarios where missing a true positive instance can lead to serious implications, such as in emergency alert systems or disease detection. A high recall indicates that the model successfully identifies most relevant cases, although it may result in more false positives [17].

$$Recall = \frac{TP}{TP+FN} \tag{11}$$

The F1-Score offers a comprehensive assessment by merging precision and recall into one harmonic average. It is particularly useful when the dataset is imbalanced and both false positives and false negatives need to be minimized. A high F1-Score suggests that the model maintains a good trade-off between identifying relevant instances and avoiding incorrect positive predictions [17].

$$F1 - Score = \frac{2\ x\ Recall\ x\ Precision}{Recall + Precision} \tag{12}$$

Execution time measures how long a model takes to classify 10,430 test tweets, recorded in seconds using the time module in Python. These measurements focus on the inference phase on the test data, with execution time also evaluated separately on the training data for comparison. In addition, the total time for the inference process on the combined training and test data is calculated to provide a practical estimate of end-to-end performance in real-world scenarios. To enhance reproducibility and contextualize the speed results, all experiments were conducted in a consistent cloud-based environment (Google Colab Pro) equipped with a single NVIDIA A100 GPU with 40 GB of VRAM. All models were executed under identical computational settings to ensure fair runtime comparisons. The results show that the models can complete the classification of the test data in under 1 second, which is considered ideal for supporting fast decision-making in real-time applications such as live social media monitoring or fraud detection [40].

4. Results and Discussion

4.1. Data examination

During the data collection stage, a technique called crawling was used to retrieve data from X platform. This technique utilized the X platform API with the help of the Python programming language. The data collected in this study consists of 10,500 tweets containing global comments about forest fires, gathered between January 1, 2024, and January 1, 2025. The tweets were specifically filtered using the English-language keywords "forest fire" and "wildfire". To prepare the collected tweets for sentiment classification, several preprocessing steps were applied to ensure the data was clean and suitable for analysis. These steps included case folding to convert all text to lowercase, removal of punctuation and irrelevant characters, elimination of stopwords to reduce noise, and tokenization to break down the text into individual terms. The following results of the pre-processing stage are shown in table 4.

Table 4. Results of The Pre-processing Stage

Processed	Result
Lower Text	opinion: a fire begins to burn in a forest. all you have is a flying air-vacuum drone made of 100% quartz + steel that smoke and fire enter through stainless steel hoop then laser-spew that smoke at the fire to suffocate its fuel of oxygen. a fire can't burn without oxygen. ???
Data Cleaning	opinion a fire begins to burn in a forest all you have is a flying airvacuum drone made of quartz steel that smoke and fire enter through stainless steel hoop then laserspew that smoke at the fire to suffocate its fuel of oxygen a fire can't burn without oxygen
Tokenization	['opinion', 'a', 'fire', 'begins', 'to', 'burn', 'in', 'a', 'forest', 'all', 'you', 'have', 'is', 'a', 'flying', 'airvacuum', 'drone', 'made', 'of', 'quartz', 'steel', 'that', 'smoke', 'and', 'fire', 'enter', 'through', 'stainless', 'steel', 'hoop', 'then', 'laserspew', 'that', 'smoke', 'at', 'the', 'fire', 'to', 'suffocate', 'its', 'fuel', 'of', 'oxygen', 'a', 'fire', 'can', 't', 'burn', 'without', 'oxygen']

 ISSN 2723-6471

Stopword	['opinion', 'fire', 'begins', 'burn', 'forest', 'flying', 'airvacuum', 'drone', 'made', 'quartz', 'steel', 'smoke', 'fire', 'enter', 'stainless', 'steel', 'hoop', 'laserspew', 'smoke', 'fire', 'suffocate', 'fuel', 'oxygen', 'fire', 'burn', 'without', 'oxygen']
Lemmatization	['opinion', 'fire', 'begin', 'burn', 'forest', 'flying', 'airvacuum', 'drone', 'made', 'quartz', 'steel', 'smoke', 'fire', 'enter', 'stainless', 'steel', 'hoop', 'laserspew', 'smoke', 'fire', 'suffocate', 'fuel', 'oxygen', 'fire', 'burn', 'without', 'oxygen']
Remove Words with Length < 3.	['opinion', 'fire', 'begin', 'burn', 'forest', 'flying', 'airvacuum', 'drone', 'made', 'quartz', 'steel', 'smoke', 'fire', 'enter', 'stainless', 'steel', 'hoop', 'laserspew', 'smoke', 'fire', 'suffocate', 'fuel', 'oxygen', 'fire', 'burn', 'without', 'oxygen']

After preprocessing, the dataset was reduced to 10,430 cleaned tweets. Prior to model training, sentiment labels had been assigned to each tweet using the VADER tool, which classifies text based on polarity scores. To ensure label reliability, the output generated by VADER was reviewed and validated by a language expert with a doctoral qualification in applied linguistics. Table 4 presents a sample of the sentiment labeling results obtained through VADER. An example of the sentiment labeling results using VADER is presented in table 5.

Table 5. Example of the Sentiment Labeling

Final Text	Labeling
opinion fire begins burn forest fly airvacuum drone made quartz steel smoke fire enter stainless steel hoop laserspew smoke fire suffocates fuel oxygen fire burn without oxygen	negative
look like forest fire beautiful	positive
nobody knows sure let see either trump want fire politics forever	positive
there forest fire winter toronto	negative

Table 5 shows the results of the data labeling process using VADER. Sentiment is divided into three categories: positive, negative, and neutral. After data annotation, the textual data was transformed into numerical form using the TF-IDF method, which calculates the importance of words relative to their frequency across documents, allowing the machine learning model to process tweet texts as meaningful numerical vectors, as illustrated in figure 3.

	Never	Statewide	Strengthen	Unveils	Wildfire	bias	birthday	she	\
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

	slipping	ult
0	0.0	0.0
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0

Figure 3. Example of Sparse TF-IDF Feature Values After SMOTE Oversampling

Figure 3 presents the output of the TF-IDF transformation applied to a collection of tweets. Each column corresponds to a unique word, while each row represents an individual tweet. The numerical values indicate the importance of each word within a tweet relative to the entire corpus. This transformation enables the machine learning model to interpret textual data as structured numerical input for further analysis. After this process is successfully executed, the SMOTE is applied to address class imbalance in the dataset. This technique generates synthetic examples for the minority class to ensure a more balanced distribution. All values are 0.0 because the selected words do not appear in those specific tweet rows. In TF-IDF, a word only gets a non-zero value if it exists in a document. Since each tweet contains only a few words while the feature space contains many terms, most values become zero. This is normal because TF-IDF produces a sparse matrix and does not indicate an error.

As a result, the model can process the data more fairly and accurately, with the visualization of SMOTE application shown in figure 4. Figure 4 illustrates the sentiment distribution before and after applying SMOTE. Initially, the dataset was imbalanced, with 4,020 negative, 1,449 neutral, and 2,875 positive instances. After applying SMOTE, each sentiment class was balanced to approximately 4,015 instances, ensuring equal representation of negative, neutral, and positive sentiments. This balancing process helps prevent bias during model training and improves classification performance across all classes. The final process at this stage is to divide the data into training and testing subsets using an 80:20 ratio to ensure sufficient data for model training while retaining some for evaluation.

ISSN 2723-6471

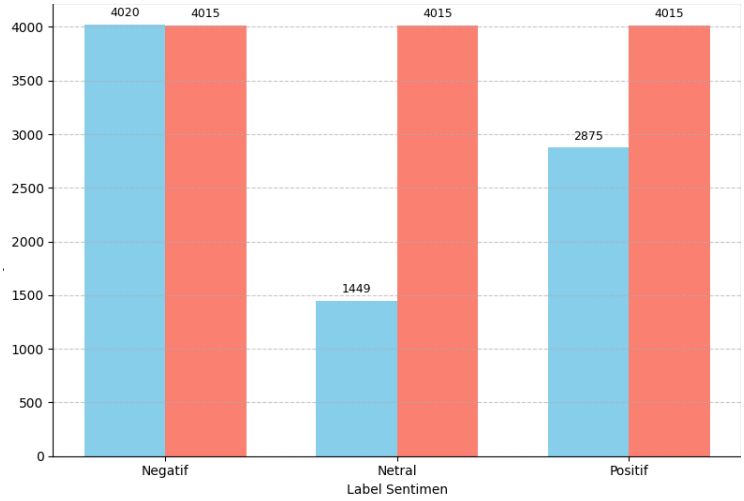

Figure 4. Sentiment Class Distribution Before and After Applying SMOTE Oversampling to Balance the Dataset

This study employed two types of datasets, namely a training set that was balanced using SMOTE to address class imbalance and enhance model learning, and a test set that remained unaltered to ensure an unbiased evaluation of model performance. Following data cleaning and preprocessing, feature selection was conducted using two swarm intelligence algorithms PSO and FA. The selected features were then used to train classification models, including SVM, Naïve Bayes, and KNN. The test data was processed directly by the classifiers without SMOTE to reflect real-world scenarios. Among all tested models, the SVM consistently achieved the superior performance, characterized by the highest accuracy along with consistent precision, recall, and F1-score metrics, which was observed in the SVM algorithm. The following is a graphic display of the comparison of the accuracy of each model in figure 5.

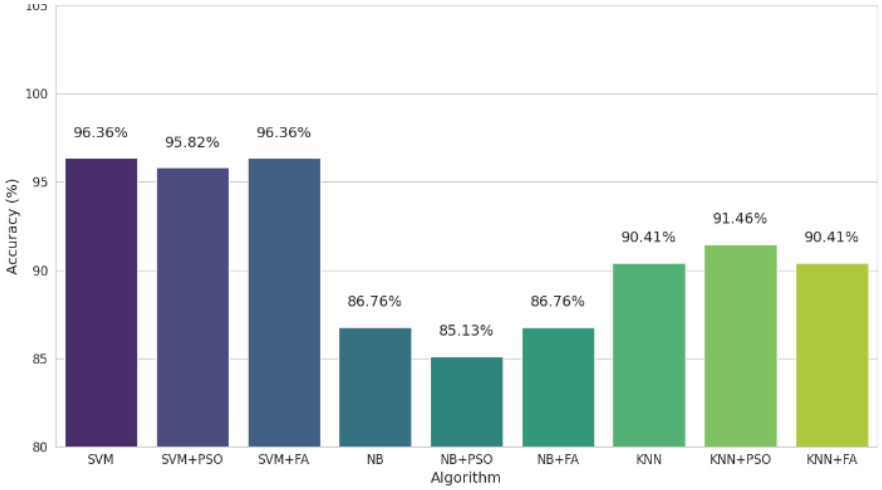

Figure 5. Comparison of Accuracy of All Models

Figure 5 presents a comparison of model accuracy with and without the application of feature selection techniques. The Support Vector Machine consistently achieved the highest accuracy, reaching 96.36% both with and without the integration of the Firefly Algorithm, demonstrating strong robustness and stability. In contrast, Naïve Bayes recorded the lowest accuracy, decreasing to 85.13% when optimized with Particle Swarm Optimization, while remaining steady at 86.76% with and without the Firefly Algorithm. Meanwhile, the K-Nearest Neighbor algorithm exhibited relatively consistent performance, with a slight improvement to 91.46% when Particle Swarm Optimization was applied. In terms of execution time, Firefly Algorithm significantly reduced processing time for all models. Support Vector Machine dropped from 31.955 to 0.0064 seconds, Naïve Bayes from 331.044 to under 1 second, and K-Nearest Neighbor from 639.598 to under 1 second. This highlights that feature selection not only maintains accuracy but also greatly enhances efficiency, making models more suitable for real-time or large-scale use. To further substantiate these findings, the confusion matrix provides a detailed breakdown of Support Vector Machine classification performance on the testing dataset. The matrix indicates a high number of correctly classified instances across all sentiment categories, particularly

ISSN 2723-6471

within the negative and positive classes. These results reaffirm the model's high accuracy and the beneficial impact of feature selection on classification precision, as illustrated in figure 6.

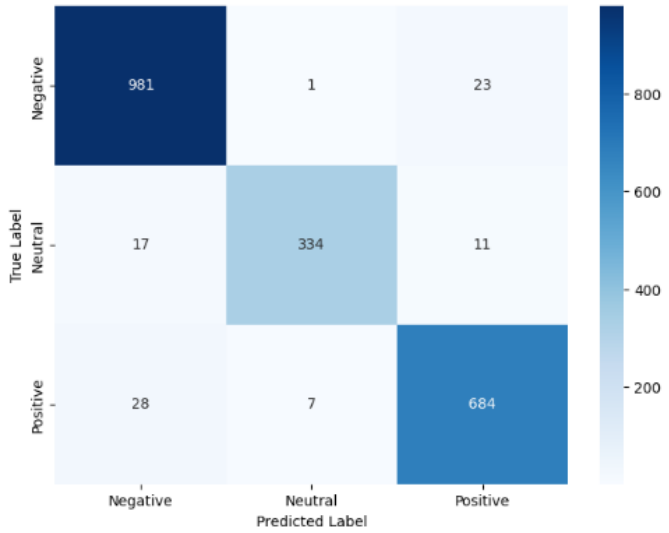

Figure 6. Confusion Matrix of Best Model

Figure 6 presents the confusion matrix for the testing data, illustrating the model's performance in classifying sentiment into negative, neutral, and positive categories. The model correctly classified 981 negative, 334 neutral, and 684 positive instances. Misclassifications are relatively minimal, with the most notable errors involving 28 positive samples predicted as negative and 23 negative samples predicted as positive. Overall, the matrix demonstrates strong classification accuracy, particularly for the negative and positive classes, indicating that the model effectively distinguishes sentiment categories with limited overlap. This result is further supported by figure 7, which shows the number of predictions per class across different models, highlighting the consistency and superiority of Support Vector Machine in handling sentiment classification.

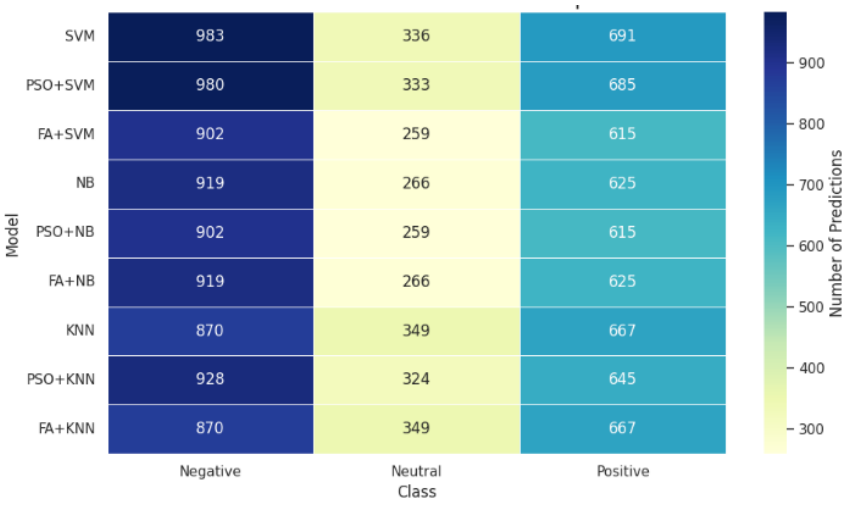

Figure 7. Number of Predictions per Class for Each Model

Figure 7 shows the prediction count for each sentiment class produced by all models. The Support Vector Machine performed best in the Negative class, with 983 correct predictions, and in the Positive class, with 691, while maintaining 336 correct predictions in the Neutral class. When combined with Particle Swarm Optimization, the prediction pattern remained similar. However, using the Firefly Algorithm led to lower counts in several categories, particularly in the Neutral and Positive classes. Naïve Bayes showed a balanced distribution with 919 predictions for Negative and 625 for Positive, but adding optimization methods did not significantly improve class-wise predictions. K-Nearest Neighbor had the highest Neutral-class predictions with 349. With the Firefly Algorithm, KNN exhibits improved class-wise

ISSN 2723-6471

prediction counts, particularly for the Negative and Positive sentiments. Overall, each model showed distinct strengths in classifying specific sentiments, as illustrated in figure 8.

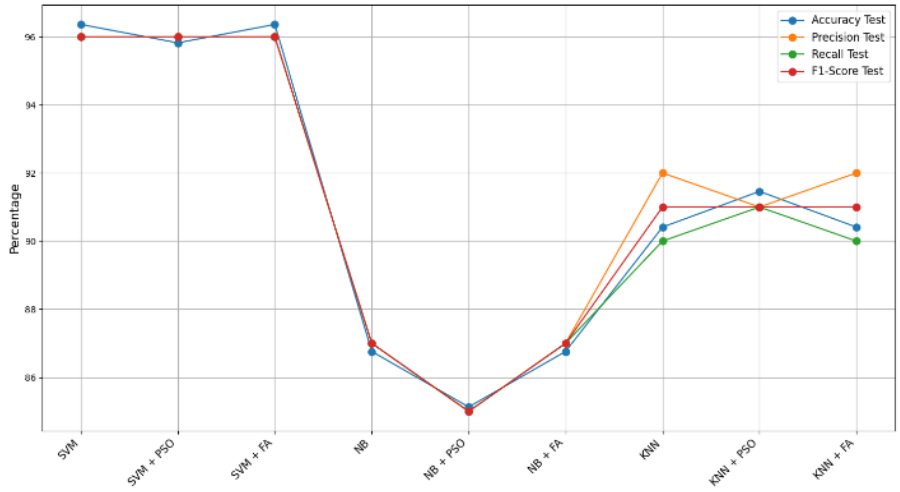

Figure 8. Model Performance Comparasion

Figure 8 illustrates a comprehensive comparison of model performance on the testing dataset, focusing on evaluation metrics: accuracy, precision, recall, and F1-score. Support Vector Machine stands out as the most robust classifier, consistently achieving high scores across all metrics regardless of whether feature selection was applied. Support Vector Machine integrated with the Firefly Algorithm maintained strong and stable performance, highlighting the effectiveness of swarm intelligence-based feature selection in enhancing classification outcomes. On the other hand, Naïve Bayes produced the lowest scores, with minimal gains observed even after applying optimization techniques. Meanwhile, K-Nearest Neighbor demonstrated notable improvements, especially when combined with Particle Swarm Optimization, which led to increased precision and F1-score. The detailed performance metrics corresponding to each model configuration are provided in table 6.

Table 6. Model Performance Summary

Algorithm	Feature Selection	Accuracy Train (%)	Accuracy Test (%)	Precision Test (%)	Recall Test (%)	F1-Score Test (%)	Execution Time (s)
SVM	-	100.0	96.36	96.00	96.00	96.00	31.955
	PSO	100.0	95.82	96.00	96.00	96.00	26.7769
	FA	100.0	96.36	96.00	96.00	96.00	0.0064
Naïve Bayes	-	88.21	86.76	87.00	87.00	87.00	331.044
	PSO	87.17	85.13	85.00	85.00	85.00	0.0093
	FA	88.23	86.76	87.00	87.00	87.00	0.0071
KNN	-	100.0	90.41	92.00	90.00	91.00	639.598
	PSO	100.0	91.46	91.00	91.00	91.00	0.9398
	FA	100.0	90.41	92.00	90.00	91.00	0.7535

Table 6 presents the evaluation of model performance based on training and testing accuracy, precision, recall, F1-score, and execution time. Among all models, Support Vector Machine consistently performed best, achieving 100% training accuracy and 96.36% testing accuracy in both baseline and Firefly Algorithm versions. Although perfect training accuracy can indicate overfitting, the use of the SMOTE balanced class distribution by generating synthetic samples for minority classes, improving model generalization.

Feature selection methods based on swarm intelligence, including Particle Swarm Optimization and Firefly Algorithm, improved both accuracy and computational efficiency. The Support Vector Machine model combined with Particle Swarm Optimization maintained a similar test accuracy of 95.82% while reducing execution time from 31.955 s to 26.7769 s. Meanwhile, the Firefly Algorithm-based version achieved even higher efficiency, with an execution time of only 0.0064 s without reducing accuracy. Similar patterns were also observed in K-Nearest Neighbor and Naïve Bayes,

where optimization preserved model accuracy while significantly reducing computation time. These findings indicate that swarm-based feature selection is effective in optimizing both classification accuracy and processing speed.

It is important to note that several models achieved 100% training accuracy. While this indicates that the classifiers can fit the balanced training data effectively, such perfect training performance may also signal potential overfitting, particularly given the high-dimensional TF-IDF feature space. Therefore, the main conclusions of this study are derived from the testing results, which better reflect generalization performance in unseen data. Future work may further strengthen robustness by incorporating cross-validation and additional generalization analyses.

4.2. Discussion

This study highlights the important role of feature selection in improving both the effectiveness and efficiency of sentiment classification models for forest fire-related tweets. Specifically, this study examines how swarm intelligence-based optimizers, namely Particle Swarm Optimization and Firefly Algorithm, affect the performance of three widely used classifiers: Support Vector Machine, Naïve Bayes, and K-Nearest Neighbor. As summarized in table 7, the integration of PSO and FA with these classifiers produces varying performance gains, indicating that the impact of optimization depends on the underlying learning mechanism of each model and the high-dimensional nature of TF-IDF features. Furthermore, table 7 provides a broader comparison that contextualizes the findings against related studies, reinforcing the relevance of swarm-based feature selection for improving sentiment classification in disaster-related social media data.

From a cost–benefit perspective, the findings suggest that swarm intelligence-based feature selection offers substantial efficiency gains while largely preserving competitive accuracy. In several cases, the optimized models achieve comparable test performance to their baseline counterparts while drastically reducing execution time, indicating that the primary benefit of PSO and FA lies in mitigating the computational burden associated with high-dimensional TF-IDF features. This trade-off is particularly advantageous for real-time or large-scale applications such as live public sentiment monitoring during disaster events. However, for use cases that prioritize simplicity or interpretability over speed, conventional classifiers, especially Naïve Bayes, remain practical options. In such contexts, swarm-based feature selection can be treated as an optional enhancement rather than a mandatory component of the sentiment analysis pipeline.

Table 7. Modeling Accuracy Comparison

Model	Feature Selection	Accuracy (%)	
		This Study	Previous Study
SVM	PSO	95.82	98.20 [19]
Naïve Bayes	PSO	85.13	93.10 [24]
KNN	PSO	91.75	90.17 [16]
SVM	FA	96.36	89.10 [18]

Table 7 presents a comparative analysis of classification model performance after applying PSO as a feature selection technique, with results from this study benchmarked against prior research. The SVM model combined with PSO achieved an accuracy of 95.82%, which is slightly lower than the 98.20% reported in a previous study [19] yet still indicates strong performance under the dataset and pipeline used in this work. Naïve Bayes with PSO reached 85.13%, lower than the 93.10% reported in earlier work [24], suggesting that the effectiveness of PSO may vary depending on classifier characteristics and the specific properties of the dataset. Meanwhile, KNN combined with PSO achieved 91.75%, slightly outperforming the 90.17% in prior studies [16], highlighting its competitive potential. It is important to note, however, that direct cross-study comparisons should be interpreted cautiously due to possible differences in dataset scope and size, labeling strategy, preprocessing steps, feature representation, imbalance handling, parameter settings, and evaluation protocols. Therefore, table 7 is intended to contextualize our results and to show that PSO-based feature selection remains competitive within the standardized experimental setting of this study.

These findings collectively highlight the strengths and limitations of using PSO across different classifiers. Although SVM consistently delivers high accuracy, the relatively lower performance of Naïve Bayes suggests that classifier selection plays a critical role in maximizing the benefits of feature selection techniques. Furthermore, this study's broader contribution lies in its use of both PSO and the FA, offering a more comprehensive evaluation than previous works that typically focus on a single method. By systematically comparing multiple algorithms using a standardized preprocessing pipeline and consistent dataset, the study provides robust insights into the trade-offs between accuracy, model complexity, and suitability for real-time sentiment analysis. These results underscore the need for careful

alignment between feature selection methods and classification algorithms to achieve optimal performance in environmental discourse analysis on social media platforms.

5. Conclusion

This study analyzes global forest fire sentiment using English tweets from the X platform through a KDD framework involving data cleaning, VADER labeling, TF-IDF transformation, and SMOTE balancing. The performance of Support Vector Machine, Naïve Bayes, and K-Nearest Neighbors was evaluated with and without swarm intelligence-based feature selection using Particle Swarm Optimization and the Firefly Algorithm. The Support Vector Machine-Firefly Algorithm model achieved the highest accuracy of 96.36% with a very fast execution time of 0.0064 seconds. KNN optimized with PSO improved its accuracy to 91.75% while reducing execution time to 0.93 seconds. Although Naïve Bayes achieved a lower maximum accuracy of 86.76%, it remained the fastest model with execution time under 0.01 seconds, making it suitable for scenarios where speed is prioritized over precision. Overall, swarm intelligence-based feature selection demonstrates strong potential to enhance both the effectiveness and efficiency of conventional sentiment classifiers. This study has several limitations. First, sentiment polarity was derived using VADER, which may be less reliable for sarcasm, irony, or context-heavy expressions. Second, disaster-related discussions often contain domain-specific sentiment cues that may not be fully represented in a general-purpose lexicon. Third, cultural and regional variations in language use may also influence polarity interpretation even within English tweets. Future work should expand the dataset across languages and time to assess generalizability and temporal robustness. Label reliability can be strengthened by incorporating a manually annotated subset, exploring domain-adapted sentiment resources for disaster discourse, and integrating sarcasm detection to better handle implicit and nuanced expressions. In addition, further studies may explore other swarm algorithms such as Salp Swarm Algorithm (SSA), Cat Swarm Optimization (CSO), and Grey Wolf Optimizer (GWO) as these methods offer different exploration–exploitation behaviors compared to PSO and FA, which may yield more robust feature subsets in high-dimensional TF-IDF settings. Future research should also test various SVM kernels and compare alternative imbalance-handling strategies, including Random Over-Sampling (ROS) and Adaptive Synthetic Sampling (ADASYN), applied only to the training set. Finally, the optimized models should be evaluated within real-time public monitoring or policy-support systems to enable faster and more reliable crisis response.

6.1. Author Contributions

Conceptualization: W.S.S and D.A.K.; Methodology: W.S.S and D.A.K.; Software: W.S.S.; Validation: W.S.S. and D.A.K. Formal Analysis: W.S.S. and D.A.K.; Investigation: W.S.S.; Resources: D.A.K.; Data Curation: W.S.S.; Writing Original Draft Preparation: W.S.S.; Writing Review and Editing: W.S.S and D.A.K.; Visualization: W.S.S.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received financial support for the research, authorship, and/or publication of this article from Universitas Multimedia Nusantara.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Suhardono, "Human activities and forest fires in Indonesia: An analysis of the Bromo incident and implications for conservation tourism," *Trees, Forests and People*, vol. 15, no. March, Art. no. 100509, pp. 1–12, Mar. 2024, doi: 10.1016/j.tfp.2024.100509.

[2] D. Fan, M. Wang, T. Liang, H. He, Y. Zeng, and B. Fu, "Estimation and trend analysis of carbon emissions from forest fires in mainland China from 2011 to 2021," *Ecological Informatics*, vol. 81, no. July, Art. no. 102572, pp. 1–16, Jul. 2024, doi: 10.1016/j.ecoinf.2024.102572.

[3] G. Tripathy and A. Sharaff, "AEGA: Enhanced feature selection based on ANOVA and extended genetic algorithm for online customer review analysis," *Journal of Supercomputing*, vol. 79, no. 12, pp. 13180–13209, Mar. 2023, doi: 10.1007/s11227-023-05179-2.

[4] M. Hadni and H. Hjiaj, "New model of feature selection based chaotic firefly algorithm for Arabic text categorization," *The International Arab Journal of Information Technology*, vol. 20, no. 3A, pp. 461–468, May 2023, doi: 10.34028/iajit/20/3A/3.

[5] A. Z. Ahmed and M. Rodríguez Díaz, "A methodology for machine-learning content analysis to define the key labels in the titles of online customer reviews with the rating evaluation," *Sustainability*, vol. 14, no. 15, Art. no. 9183, pp. 1–31, Jul. 2022, doi: 10.3390/su14159183.

[6] D. A. Putri, D. A. Kristiyanti, E. Indrayuni, A. Nurhadi, and D. R. Hadinata, "Comparison of Naive Bayes algorithm and Support Vector Machine using PSO feature selection for sentiment analysis on e-wallet review," *Journal of Physics: Conference Series*, vol. 1641, no. 1, Art. no. 012085, pp. 1-7, Nov. 2020, doi: 10.1088/1742-6596/1641/1/012085.

[7] G. Srivastava, V. Singh, and S. Kumar, "Hybrid model for sentiment analysis combination of PSO, genetic algorithm and voting classification," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 2, pp. 1151–1161, Aug. 2024, doi: 10.11591/ijeecs.v35.i2.pp1151-1161.

[8] H. Ullah , "Comparative study for machine learning classifier recommendation to predict political affiliation based on online reviews," *CAAI Transactions on Intelligence Technology*, vol. 6, no. 3, pp. 251–264, May 2021, doi: 10.1049/cit2.12046.

[9] A. Sharma and U. Ghose, "Toward machine learning based binary sentiment classification of movie reviews for resource restraint language (RRL)–Hindi," *IEEE Access*, vol. 11, no. June, pp. 58546–58564, Jun. 2023, doi: 10.1109/ACCESS.2023.3283461.

[10] A. Lubis, Y. Irawan, Junadhi, and S. Defit, "Leveraging K-nearest neighbors with SMOTE and boosting techniques for data imbalance and accuracy improvement," *Journal of Applied Data Sciences*, vol. 5, no. 4, pp. 1625–1638, Dec. 2024, doi: 10.47738/jads.v5i4.343.

[11] A. Londhe and P. V. R. D. P. Rao, "Incremental learning based optimized sentiment classification using hybrid two-stage LSTM-SVM classifier," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, pp. 611–619, Jun. 2022, doi: 10.14569/IJACSA.2022.0130674.

[12] S. L. Marie-Sainte and N. Alalyani, "Firefly algorithm based feature selection for Arabic text classification," *Journal of King Saud University – Computer and Information Sciences*, vol. 32, no. 3, pp. 320–328, Mar. 2020, doi: 10.1016/j.jksuci.2018.06.004.

[13] R. Obiedat , "Sentiment analysis of customers' reviews using a hybrid evolutionary SVM-based approach in an imbalanced data distribution," *IEEE Access*, vol. 10, no. March, pp. 22260–22273, Mar. 2022, doi: 10.1109/ACCESS.2022.3149482.

[14] M. Rezapour, "Sentiment classification of skewed shoppers' reviews using machine learning techniques, examining the textual features," *Engineering Reports*, vol. 3, no. 1, pp. 1–13, Mar. 2021, doi: 10.1002/eng2.12280.

[15] D. Widyawati and A. Faradibah, "Comparison analysis of classification model performance in lung cancer prediction using Decision Tree, Naive Bayes, and Support Vector Machine," *Indonesian Journal of Data and Science*, vol. 4, no. 2, pp. 80–89, Jul. 2023, doi: 10.56705/ijodas.v4i2.76.

[16] T. Sinnasamy and N. N. A. Sjaif, "Sentiment analysis using term based method for customers' reviews in Amazon product," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 7, pp. 685–691, Jul. 2022, doi: 10.14569/IJACSA.2022.0130780.

[17] D. A. Kristiyanti, S. A. Sanjaya, V. C. Tjokro, and J. Suhali, "Dealing imbalance dataset problem in sentiment analysis of recession in Indonesia," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 2058–2070, Jun. 2024, doi: 10.11591/ijai.v13.i2.pp2060-2072.

[18] D. A. Anggoro and S. S. Mukti, "Performance comparison of grid search and random search methods for hyperparameter tuning in Extreme Gradient Boosting algorithm to predict chronic kidney failure," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 6, pp. 198–207, Aug. 2021, doi: 10.22266/ijies2021.1231.19.

[19] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *IEEE Access*, vol. 10, no. January, pp. 10031–10061, Jan. 2022, doi: 10.1109/ACCESS.2022.3142859.

[20] D. A. Putri, D. A. Kristiyanti, E. Indrayuni, A. Nurhadi, and D. A. Muthia, "Mining Twitter data on COVID-19 for sentiment analysis using SVM algorithm," in *AIP Conference Proceedings*. Jakarta, Indonesia: AIP Publishing, vol. 2023, no. May, pp. 1–6, 2023, doi: 10.1063/5.0128833.

[21] G. Nagarajan and L. D. Dhinesh Babu, "A hybrid feature selection model based on improved squirrel search algorithm and rank aggregation using fuzzy techniques for biomedical data classification," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 10, no. 1, pp. 1–29, Jun. 2021, doi: 10.1007/s13721-021-00313-7.

[22] Arpita, P. Kumar, and K. Garg, "Data cleaning of raw tweets for sentiment analysis," in *Proc. Indo-Taiwan 2nd International Conference on Computing, Analytics and Networks (ICAN)*. Rajpura, India: IEEE, vol. 2020, no. Sep., pp. 273–276, 2020, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181326.

[23] E. I. Onal, S. Tekeli-Yeşil, and N. Okay, "The use of Twitter by official institutions in disaster risk communication and resilience," *Journal of Emergency Management and Disaster Communications*, vol. 3, no. 1, pp. 25–40, Apr. 2022, doi: 10.1142/s2689980922500087.

[24] T. Ho, H. M. Bui, and T. K. Phung, "A hybrid model for aspect-based sentiment analysis on customer feedback: Research on the mobile commerce sector in Vietnam," *International Journal of Advances in Intelligent Informatics*, vol. 9, no. 2, pp. 273–285, Jul. 2023, doi: 10.26555/ijain.v9i2.976.

[25] L. K. Ramasamy, S. Kadry, and S. Lim, "Selection of optimal hyper-parameter values of Support Vector Machine for sentiment analysis tasks using nature-inspired optimization methods," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 290–298, 2021, doi: 10.11591/eei.v10i1.2098.

[26] F. Akbarian and F. Z. Boroujeni, "An improved feature selection method for sentiments analysis in social networks," in *Proc. 10th International Conference on Computer and Knowledge Engineering (ICCKE)*. Mashhad, Iran: IEEE, vol. 2020, no. Oct., pp. 181–186, 2020, doi: 10.1109/ICCKE50421.2020.9303710.

[27] K. Rezaei and H. Rezaei, "An improved firefly algorithm for numerical optimization problems and its application in constrained optimization," *Engineering with Computers*, vol. 38, no. 4, pp. 3793–3813, Aug. 2022, doi: 10.1007/s00366-021-01412-9.

[28] D. Elreedy and A. F. Atiya, "A comprehensive analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance," *Information Sciences*, vol. 505, no. December, pp. 32–64, Dec. 2019, doi: 10.1016/j.ins.2019.07.070.

[29] M. A. Virgananda, I. Budi, and R. R. Suryono, "Purchase intention and sentiment analysis on Twitter related to social commerce," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 7, pp. 543–550, 2023, doi: 10.14569/IJACSA.2023.0140760.

[30] P. Mehta, S. Pandya, and K. Kotecha, "Harvesting social media sentiment analysis to enhance stock market prediction using deep learning," *PeerJ Computer Science*, vol. 7, no. April, pp. 1-21, 2021, doi: 10.7717/peerj-cs.476.

[31] B. N. Hiremath and M. M. Patil, "Enhancing optimized personalized therapy in clinical decision support system using natural language processing," *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 6, pp. 2840–2848, Jun. 2022, doi: 10.1016/j.jksuci.2020.03.006.

[32] M. I. A. Latiffi, M. R. Yaakub, and I. S. Ahmad, "Flower pollination algorithm for feature selection in tweets sentiment analysis," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, pp. 429–436, 2022, doi: 10.14569/IJACSA.2022.0130551.

[33] S. Jain and A. Sinha, "Identification of influential users on Twitter: A novel weighted correlated influence measure for COVID-19," *Chaos, Solitons & Fractals*, vol. 139, Art. no. 110037, pp. 1-8, Jun. 2020, doi: 10.1016/j.chaos.2020.110037.

[34] F. H. Fadel and S. F. Behadili, "A comparative study for supervised learning algorithms to analyze sentiment tweets," *Iraqi Journal of Science*, vol. 63, no. 6, pp. 2712–2724, Jun. 2022, doi: 10.24996/ijs.2022.63.6.36.

[35] J. Rumbut , "Harmonizing wearable biosensor data streams to test polysubstance detection," in *Proc. International Conference on Computing, Networking and Communications (ICNC)*. North Dartmouth, MA, USA: IEEE, vol. 2020, no. Feb., pp. 445–449, 2020, doi: 10.1109/ICNC47757.2020.9049684.

[36] N. S. Mohd Nafis and S. Awang, "An enhanced hybrid feature selection technique using term frequency–inverse document frequency and Support Vector Machine–recursive feature elimination for sentiment classification," *IEEE Access*, vol. 9, no. April, pp. 52177–52192, Apr. 2021, doi: 10.1109/ACCESS.2021.3069001.

[37] J. A. Cummins and V. E. Nambudiri, "Natural language processing: A window to understanding skincare trends," *International Journal of Medical Informatics*, vol. 160, no. April, Art. no. 104705, pp. 1-8, Apr. 2022, doi: 10.1016/j.ijmedinf.2022.104705.

[38] M. A. Nulhakim, Y. Sibaroni, K. Muhammad, and N. Ku, "Geospatial sentiment analysis using Twitter data on natural disasters in Indonesia with Support Vector Machine," *International Journal on ICT*, vol. 10, no. 2, pp. 242–258, Dec. 2024, doi: 10.21108/ijoict.v10i2.1032.

[39] V. Pandi, P. Nithiyanandam, S. Manickavasagam, I. M. Meerasha, R. Jaganathan, and M. K. Balasubramanian, "A comprehensive analysis of consumer decisions on Twitter dataset using machine learning algorithms," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 3, pp. 1085–1093, Sep. 2022, doi: 10.11591/ijai.v11.i3.pp1085-1093.

[40] A. Alarifi, A. Tolba, Z. Al-Makhadmeh, and W. Said, "A big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks," *Journal of Supercomputing*, vol. 76, no. 6, pp. 4414–4429, Jun. 2020, doi: 10.1007/s11227-018-2398-2.