Revisiting Support Vector Machines: A Distance-Based Alternative to Deep Learning for Efficient Text Classification

Siew Teng Koh[1], Anbuselvan Sangodiah[2], Norazira Binti A Jalil[3],
Jafhate Edward[4], Nur Fatin Liyana Binti Mohd Rosely[5]
Faculty of Integrated Life Science, Quest International University, Ipoh, Perak, 30250, Malaysia [1]
Faculty of Innovation and Technology, Taylor's University, Subang Jaya, 47500, Selangor, Malaysia [2, 4, 5]
Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar, 31900, Perak Malaysia [3]

Abstract*—Support Vector Machines (SVMs) remain competitive in text classification, sometimes achieving comparable performance with the deep learning approach, due to their strong generalization ability and robustness to overfitting. However, their strong classification performance relies heavily on the selection of kernel functions and parameters, resulting in substantial hyperparameter tuning. A previous study has proposed Euclidean-SVM, which modifies the SVM decision mechanism by replacing the optimal separating hyperplane with a distance-based decision rule. This proposed approach reported reduced dependency on kernel functions and regularization parameters, resulting in robust performance with lower sensitivity to hyperparameter changes. Nevertheless, Euclidean-SVM only investigates Euclidean distance; other distance metrics that may achieve comparable performance remain unexplored. This study aims to evaluate the effectiveness of multiple distance metrics as an alternative decision function in the distance-based SVM framework for text classification. The distances, including Euclidean distance, Manhattan distance, Chebyshev distance, Cosine distance, and Minkowski distance, were investigated. The experimental results demonstrate that Euclidean and Cosine distances achieve stable and competitive classification performance across a wide range of hyperparameter configurations, reaching an accuracy of approximately 84-97% across the evaluated datasets. In contrast, the remaining distances, including Manhattan, Chebyshev, and Minkowski, exhibit significantly lower performance, reaching an accuracy between 14 and 71%, indicating the discriminative power of these distances is lower. A preliminary comparison with the deep learning model Long Short-Term Memory (LSTM) further shows that the distance-based SVM, including Euclidean and Cosine-based SVM, achieves higher performance and greater stability. These findings suggest that Euclidean and Cosine distances enhance the robustness of SVM-based text classification, while reducing the need for extensive hyperparameter tuning, making them suitable for resource-constrained environments compared to deep learning.

Keywords*—SVM; text classification; Euclidean-SVM; Cosine distance; Manhattan distance; Chebyshev distance; Minkowski distance; LSTM

I. Introduction

Text classification is a fundamental task in Natural Language Processing (NLP), which is the task of automatically assigning a predefined category label to a text document. The rapid growth of digital content generated from digital platforms such as social media, blogs, and news portals has led to a vast volume of unstructured textual data. Text classification plays an important role in effectively organizing and interpreting this data by transforming the unstructured text data into structured information. As a result, text classification has become a significant task in various real-world applications, including spam detection, sentiment analysis, and topic labelling.

SVM has been widely used in text classification due to its effectiveness in handling the high-dimensional feature spaces and strong generalization capability through its margin-based optimization [1][2]. In the text classification task, extraction methods such as bag-of-words and Term Frequency-Inverse Document Frequency (TF-IDF) typically produce sparse and high-dimensional numerical vectors, where SVM performs particularly well.

Although deep learning models have dominated in many NLP tasks, a recent study shows that the traditional machine learning classifier, such as SVM with TF-IDF representation, can achieve comparable performance to fine-tuned Pretrained Language Models (PLMs) in several text classification tasks [3]. SVM also remains one of the most commonly used machine learning classifiers for text classification tasks, frequently demonstrating strong performance in many scenarios [4]. In addition to classical machine learning, SVM has also been studied in quantum computing. In [5], the authors demonstrated that the Quantum Support Vector Machine (QSVM) can be implemented with complexity logarithmic in feature dimensions and the number of training samples, potentially providing an exponential speedup over classical sampling algorithms.

Despite its effectiveness, SVM classification performance is highly dependent on the selection of the kernel function and the regularization parameter. As a result, extensive hyperparameter tuning was often required to achieve optimal classification performance, thereby increasing computational cost and limiting its applicability in resource-constrained environments.

To reduce this sensitivity, [6] proposed a modified SVM classification framework named Euclidean-SVM. In this approach, the training phase of conventional SVM was

employed to identify support vectors, while the classification phase no longer relies on the constructed hyperplane. Instead, the classification decision was made by computing the Euclidean distance between the test data and the support vectors from each class. By replacing the hyperplane-based decision mechanism with a distance-based decision mechanism, Euclidean-SVM reduces the impact of kernel choices and regularization parameters on classification performance. Experimental results reported that Euclidean-SVM achieves a comparable and more consistent classification accuracy across different kernels and regularization parameter settings. Furthermore, Euclidean-SVM has demonstrated more consistent and better classification performance compared to conventional SVM in the oil and gas pipeline monitoring system [7].

However, Euclidean-SVM only employs Euclidean distance as the similarity measure. Alternative distance metrics such as Manhattan, Cosine, Chebyshev, and Minkowski remain unexplored. These distances may capture different geometrical relationships in high-dimensional feature spaces and potentially improve efficiency and robustness.

Prior studies have emphasized the importance of distance selection within the SVM framework. For example, [8] has proposed a dissimilarity-based kernel where the different distance functions determine the kernel representations. Their approach uses several dissimilarity metrics, including Jaccard, Rogers-Tanimoto coefficient, and Sokal-Michener coefficient. The experimental results demonstrate that the choice of the distance significantly affects the classification accuracy. Similarly, [9] proposed a weighted p-norm distance t-kernel within the SVM framework, where the kernel was defined by the p-norm distance. This study demonstrates that different p-norm distance significantly affects classification performance across multiple datasets. These studies further confirm the significance of distance choice in the SVM framework. However, these studies modify the kernel to incorporate different distance measures, whereas the present study investigates the impact of different distance measures in the classification phase of the support vectors mechanism classification framework.

Motivated by this gap, this study systematically investigates the effectiveness of multiple distance measures in distance-based SVM classification frameworks to evaluate their effectiveness as alternative distance measures in text classification. Specifically, the Euclidean, Manhattan, Chebyshev, Cosine, and Minkowski distances were evaluated. The Euclidean distance was commonly used due to its simplicity and intuitiveness; it may not be optimal in sparse and high-dimensional feature spaces. Different distance metrics capture different geometric characteristics of data. For instance, Manhattan distance measures the total feature difference, Chebyshev distance measures the largest difference across all features, Cosine distance measures the angle relationship between data, while Minkowski distance measures the distance between two points controlled by a tunable parameter p.

In addition to classification accuracy, this study evaluates the robustness and consistency under varying kernel functions and regularization parameter configurations. The objectives are to determine whether the alternative distance metrics can achieve a comparable or improved robustness while reducing sensitivity to hyperparameter selection. Furthermore, to provide a comprehensive evaluation, a preliminary comparison with a deep learning model, namely Long Short-Term Memory (LSTM), was conducted. This comparison was not intended as a comprehensive evaluation of all deep learning models, but provided an initial reference point for the competitiveness of the distance-based SVM. LSTM was selected due to its effectiveness in sequential data and its popularity in text classification tasks. This comparison was to evaluate whether the distance-based SVM frameworks remain competitive with the neural network-based approaches for text classification.

The remaining part of the study was organized as follows: Section II presents the materials and methods, Section III demonstrates the results and the discussion, and Section IV illustrates the conclusion from the findings.

II. Materials and Methods

A. Methodology Workflow

This section demonstrates the methodology employed to evaluate the performance of the proposed distance-based SVM in comparison with the conventional SVM. The overall workflow was designed to ensure fair comparison, reproducibility, and consistency across all models.

Fig. 1 illustrates the overall experimental pipelines. The pipelines begin with the dataset preparation and preprocessing, followed by TF-IDF feature extraction. The processed data are then used to train a conventional SVM model, and the support vectors are learned and extracted. These support vectors are utilized by the proposed distance-based SVM to perform classification. Both models are evaluated by using accuracy and F1 score on a held-out test set.

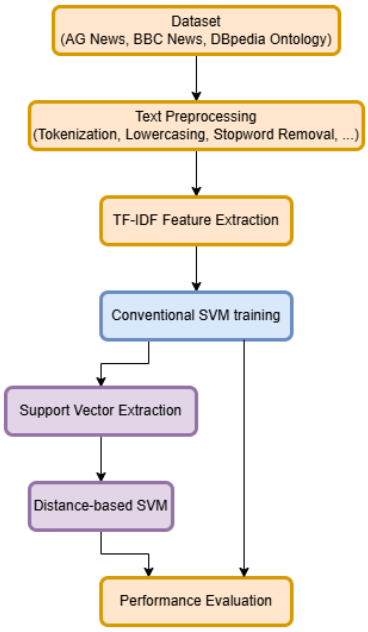

Fig. 1. Overall methodology workflow

B. Maintaining the Integrity of the Specifications

Three benchmark text classification datasets, including AG News, BBC News, and the DBpedia Ontology classification dataset, were employed to evaluate the performance and robustness of the conventional SVM and proposed distance-based SVM approach. These datasets differ in dataset size, number of classes, and text complexity, enabling a comprehensive evaluation across diverse classification scenarios. The class distributions of datasets were demonstrated in Fig. 2 to Fig. 4, which illustrate the number of training and testing sets for each class.

1) AG News dataset: AG News dataset [10] consists of news articles categorized into four classes: World, Sports, Business, and Science/Technology. Each sample consists of a news title and a short description, which were concatenated to form the textual input.

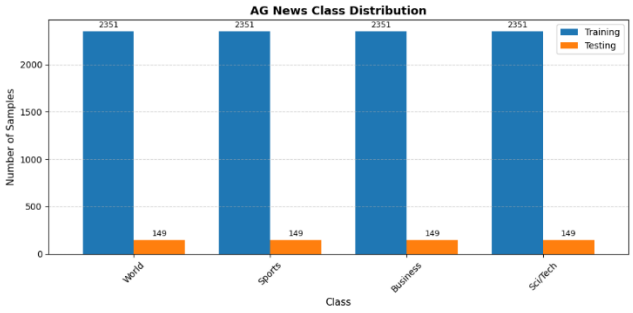

Fig. 2. AG News class distribution

As shown in Fig. 2, the dataset has a perfectly balanced class distribution. A stratified subsampling strategy was applied to construct a subset of 10,000 samples while preserving the original training-testing ratio. This balanced distribution ensures fair performance comparison without bias from class imbalance.

2) BBC News dataset: BBC News dataset [11] consists of 2225 news articles collected from the BBC News website, categorized into five topical classes: Business, Entertainment, Politics, Sport, and Technology.

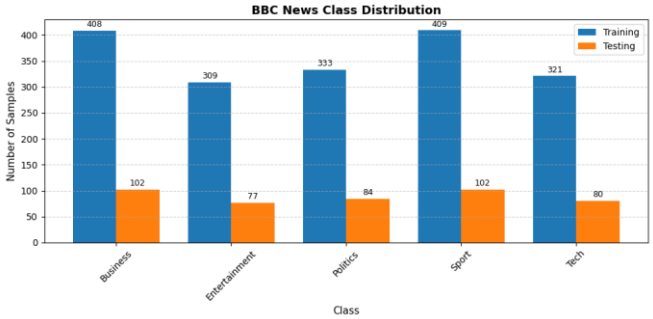

Fig. 3. BBC News class distribution

Since no official training-testing split was provided, a stratified 80/20 train-test split was employed. As shown in Fig. 3, the dataset shows mild class imbalance, reflecting a more realistic text classification scenario. The full dataset was utilized without subsampling.

3) DBpedia Ontology classification dataset: The DBpedia Ontology classification dataset is a large-scale dataset derived from structured content extracted from Wikipedia articles. Each document belongs to one of 14 ontology classes.

Each instance consists of a title and a content field, which were concatenated to form the textual input. The official training-testing split was preserved, and a class-balanced subset of approximately 10,000 samples was constructed by uniformly sampling from each class.

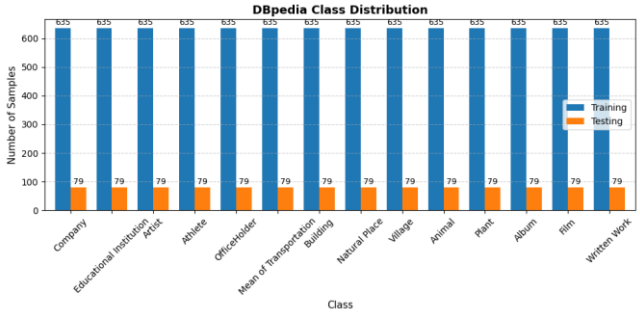

Fig. 4. DBpedia Ontology class distribution

As shown in Fig. 4, all ontology classes contain an equal number of samples. Compared to AG News and BBC News, the DBpedia Ontology classification dataset represents a more challenging task due to its larger number of classes and longer textual content.

C. Text Preprocessing

1) SVM and Distance-Based SVM: Before feature extraction, all datasets were preprocessed by using a standard preprocessing pipeline, as shown in Fig. 5, with the Natural Language Toolkit (NLTK) library. This pipeline reduces noise, standardizes textual representation, and improves feature quality for further vectorization and SVM classification.

The pipeline includes tokenization, normalization, noise removal, stopword elimination, and lemmatization to standardize textual input before feature extraction and classification. The preprocessing step includes:

- Raw Text Dataset: Input textual content from the original dataset.
- Tokenization: Split raw text into word-level tokens.
- Lowercasing: Convert all tokens to lower case to avoid duplicate feature representation.
- Remove non-alphabetic: remove punctuation, numbers, and symbols to reduce irrelevant features.
- Stopword Removal: Remove common and semantically uninformative features using NLTK's stopword list.

- Lemmatization: Reduce tokens to their base forms using WordNet lemmatizer, minimizing vocabulary size while preserving the semantic information.
- Cleaned Tokens: Rejoin processed tokens into a whitespace-separated string as input for the further text vectorization step.

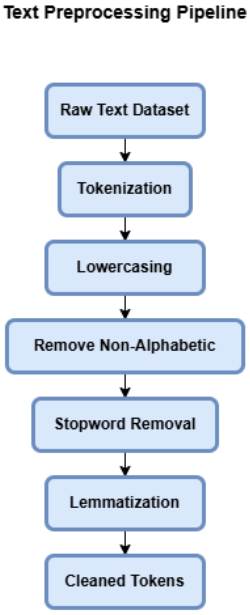

Fig. 5. Text preprocessing pipeline applied to all datasets.

2) LSTM: The raw text was converted into numerical representations by using a tokenizer. A word-level tokenizer was applied with a vocabulary size of 20,000 to ensure only the most frequent words are retained. The words that were not included in the vocabulary were mapped to an out-of-vocabulary (OOV) token.

The transformed sequences were then padded or truncated to a fixed maximum length of 500 tokens to ensure uniform length for the model input. The output of this step is a padded sequence of integer indices, which served as the input for the LSTM for the subsequent step.

D. Feature Representation

1) SVM and Distance-Based SVM: To enable the application of a machine learning algorithm, the pre-processed textual document must be transformed into a numerical representation, as a machine learning algorithm cannot directly consume raw text as input. The feature extraction was conducted by using TF-IDF, which is widely utilized in traditional text classification tasks.

TF-IDF is a statistical method that assigns weights to each word based on two components: term frequency (TF) and inverse document frequency (IDF). Term frequency determines the frequency of a term occurring in a document, while inverse document frequency penalizes the term that occurs frequently across many documents. As a result, the words that appear frequently in a specific document but rarely appear in the whole corpus will be assigned a higher weight to indicate their importance, while the common but less informative words will be assigned a lower score. This mechanism allows TF-IDF to highlight the discriminative terms that contribute more to distinguishing between classes.

Although text representation techniques, such as Word2Vec, GloVe, and contextual embeddings, such as BERT, have become more popular in NLP, TF-IDF was intentionally utilized to ensure the methodology consistency and controlled evaluation. The primary objective is to investigate the impact of different distance measures in the distance-based SVM framework. Introducing other methods like word embeddings or contextual embeddings will introduce additional complexity and confounding factors, making it more difficult to isolate the effect of distance measures.

Furthermore, TF-IDF generated high-sparse vectors, a numerical representation that well aligns with the geometric properties of SVM. This characteristic enables a clearer observation of how different distance metrics capture geometrical relationships in feature space. To ensure a fair and unbiased comparison, the TF-IDF vocabulary was learned only from the training data to prevent information leakage. The same vocabulary and weighting were then applied to transform the held-out test data. The TF-IDF was configured to limit the feature space to the 5000 most frequent terms. These settings were to balance the computational efficiency and representation capability. A moderate vocabulary size was selected to reduce sparsity while preserving sufficient important information for text classification. Since the primary objective is to evaluate the impact of different distance measures, the same TF-IDF configurations were applied in all experiments.

2) LSTM: The embedding layer was employed as the feature representation for LSTM. The embedding layer transforms the input word index into a dense 128-dimensional vector. This layer acts as a learnable feature representation module, where the words with similar semantic meaning are mapped closely in the embedding space, thus enabling the model to capture the semantic relationships between the words.

E. Model

1) Conventional SVM: The conventional SVM was employed as a baseline classifier and used as a reference for comparison with the proposed distance-based SVM variants for text classification. SVM is a supervised learning algorithm widely used for classification, known for its strong generalization and robustness to high-dimensional data. The workflow of conventional SVM was shown in Fig. 6.

The pre-processed and vectorized data served as the input for the SVM classifier. The core principle of SVM is to find an optimal separating hyperplane that maximizes the margin between different classes. The margin was defined as the distance between the hyperplane and the closest points from each class. Their closest points, known as support vectors, play a critical role in defining the decision. The support vectors represent the most informative points and are used as the

reference points in the later distance-based classification stage. This property is closely related to the margin maximization of SVM, where the support vectors were the closest point to the optimal separating hyperplane that maximize the margin between classes. Therefore, the discriminative information obtained from margin maximization was preserved when the support vectors are subsequently employed in the distance-based classification stage.

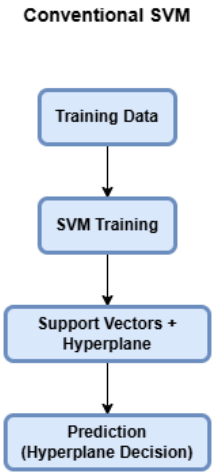

Fig. 6. Conventional SVM workflow

To enable a fair comparison of different types of data distributions, four commonly used kernel functions were employed: linear, polynomial, radial basis function (RBF), and sigmoid. The kernel function allows SVM to implicitly transform the data into a higher-dimensional feature space, enabling the data separation that the data are not linearly separable in the original space. The choice of kernel significantly affects the model's ability to capture the complex data patterns.

To ensure an unbiased evaluation, the regularization parameter was fixed in a wide range: C∈{1,10,100,1000,10000,100000}. The regularization parameter C allows SVM to control the trade-off between maximizing the margin and minimizing the classification error, thereby affecting model generalization and overfitting. The support vectors identified in the conventional SVM phase were retained and utilized in the distance-based SVM framework.

2) Distance-Based SVM: Fig. 7 demonstrates the overall workflow of the distance-based SVM. This approach shares the same training phase as conventional SVM for identifying support vectors. However, the distance-based SVM replaces the hyperplane-based decision with a distance-based mechanism. This modification does not modify the SVM optimization process but only alters the classification rule in the classification phase.

The distance-based SVM was a modified classification framework from the conventional SVM. Instead of relying on the hyperplane for classification prediction, this approach performs classification by computing the distance between the test instance and the support vectors for each class. This approach was inspired by Euclidean-SVM [6]. This study extends it by incorporating various distances instead of only the Euclidean distance.

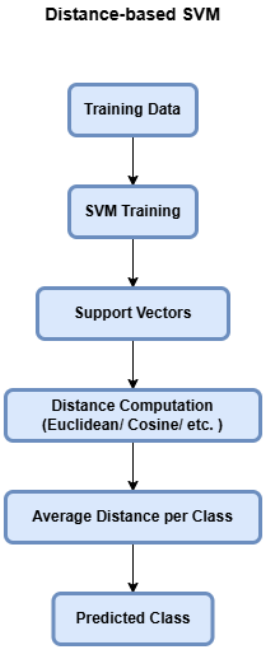

Fig. 7. Distance-based SVM workflow

In the proposed framework, the conventional SVM was first trained using the selected kernel and regularization parameter to identify the support vectors with their class labels. This training follows the standard SVM procedure and is identical to the conventional model. The aim of this training phase is solely to identify the support vectors, which are the most informative data near the decision boundary. During the classification phase, the hyperplane obtained from the training phase was not used for the classification prediction. Given a new unseen data x_{new}, the distance between the x_{new} and each support vector was computed. Subsequently, for each class y, the average distance between x_{new} and all the support vectors that belong to the class was calculated. Lastly, the predicted class, $\hat{y}$, is determined by selecting the class with the minimum average distance, as shown in Eq. (1):

$$\hat{y} = arg\ \min_{y}\left(\frac{1}{|SV_y|}\sum_{s\in SV_y} d(x_{new}, s)\right) \quad (1)$$

From Eq. (1), $|SV_y|$ denotes the number of the support vectors of class y, SV_y represents the set of support vectors of class y, while $d(x_{new}, s)$ indicates that the distance between the new instance x_{new} and the particular support vector. This decision rule assigns the new unseen data to the class that is closest to the support vectors on average. This strategy allows the classifier to decide based on the similarity relationships rather than a fixed hyperplane, which can be beneficial when the class distribution is complex and non-linear.

Unlike the Euclidean-SVM, which depends solely on the Euclidean distance, this study investigates the impact of different distance measures in the same framework. Various distance functions, including Euclidean distance, Manhattan distance, Chebyshev distance, Cosine distance, and Minkowski

distance, were investigated. Each distance defines a classifier and uses the same set of support vectors obtained from the training phase. To ensure a fair and unbiased evaluation with the conventional SVM, the distance-based SVM utilized identical kernel functions and regularization parameter configurations. All models are evaluated on a held-out test set.

3) LSTM Model: In addition to conventional SVM and distance-based SVM, a Long Short-Term Memory was employed as the deep learning baseline for text classification. LSTM is a special type of recurrent neural network that is designed to learn long-term dependencies in sequential data, making it suitable for text classification. The overall workflow of the LSTM was illustrated in Fig. 8.

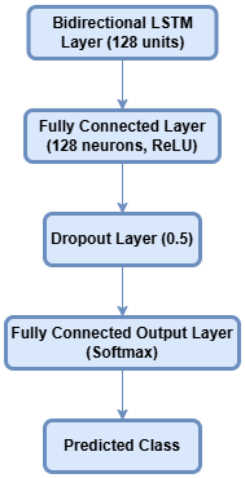

Fig. 8. LSTM workflow

The embedded sequences were passed to a bidirectional LSTM layer with 128 units. This layer processes the input sequence in both forward and backward directions, making it better at capturing the contextual information from both past and future words in the sequences. The output is then passed to a fully connected layer with 128 neurons and a ReLU activation function. This layer refines the learning representations for classification and learns the high-level features. A dropout layer with a dropout rate of 0.5 was then applied. This dropout layer helps to prevent overfitting by randomly deactivating a part of a neuron during training.

The model utilized a fully connected layer with a softmax activation function as the final layer. This layer produces a probability distribution over the target classes for multi-class classification. The model was trained by using the Adam optimizer with a learning rate of 0.001. The loss function utilized was categorical cross-entropy. Early stopping with a patience of 3 was applied to prevent overfitting.

F. Evaluation Metrics

The classification performance of the conventional SVM and distance-based SVM was evaluated by using two complementary metrics: accuracy and macro F1 score. Accuracy provides a general overview of model robustness, which measures the proportion of correctly classified instances to the total number of instances [see Eq. (2)]:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (2)$$

Macro F1 score captures the harmonic mean of precision and recall. For multi-class classification, the macro F1 score calculates the mean of the F1 score across all the classes independently, then averages them, treating all the classes evenly regardless of the class size [see Eq. (3)]:

$$F1_{macro} = \frac{F1_1 + F1_2 + \cdots + F1_N}{N} \quad (3)$$

where, N is the number of classes.

Accuracy was used as the primary metric to assess the performance comparison, while the macro F1 score provided additional insights for model robustness. Both metrics were computed on the test set for each kernel and regularization parameter configuration.

G. Experimental Setup and Evaluation Protocol

This section describes the experimental configuration used to evaluate the conventional SVM and distance-based SVM under a consistent and reproducible environment. The experiment was conducted in Python 3.13.5 by using scikit-learn 1.7.1, Numpy 2.3.2, pandas 2.3.1, and NLTK 3.9.1. To ensure reproducibility, a fixed random seed (random seed=42) was applied consistently across dataset splitting and SVM training. To ensure a fair comparison between SVM, distance-based SVM, and the LSTM model, the training and testing set was consistency between all models. The training data was used to build the model, while the held-out test set was utilized to evaluate the model's performance. To enable a fair comparison between SVM and distance-based SVM, both models:

- Identical TF-IDF numerical representation.
- Shared the same kernel function (linear, polynomial, RBF, and sigmoid) and regularization parameter ($C = 1, 10, 10^2, 10^3, 10^4, 10^5$).

To ensure a fair comparison between the SVM-based methods and LSTM, LSTM was conducted under the same dataset split and fixed random seed of 42. Final performance was evaluated on the test set by using accuracy and macro F1 score as the primary metrics.

III. Results and Discussion

A. Overall Performance Comparison Across SVM and Distance-Based SVM

This section presents the classification performance comparison of conventional SVM with each distance-based SVM variant, including Euclidean, Manhattan, Chebyshev, Cosine, and Minkowski across three text classification datasets: AG News, BBC News, and DBpedia Ontology classification dataset under various configurations. The performance was evaluated by using accuracy and F1 score, as presented in the Appendix (see Table I to Table VI) and illustrated in Fig. 9 to Fig. 14.

Fig. 9 to Fig. 11 demonstrate the accuracy of conventional SVM and distance-based SVM variants across different kernel and regularization parameter values on AG News, BBC News,

and DBpedia Ontology classification datasets. Each subfigure corresponds to a specific method, including conventional SVM and each distance-based variant. The individual curve represents a different kernel function. The horizontal axis shows the regularization parameter on a logarithmic scale, while the vertical axis demonstrates the accuracy in percentage. The exact numerical result is provided in the Appendix (see Table I to Table III).

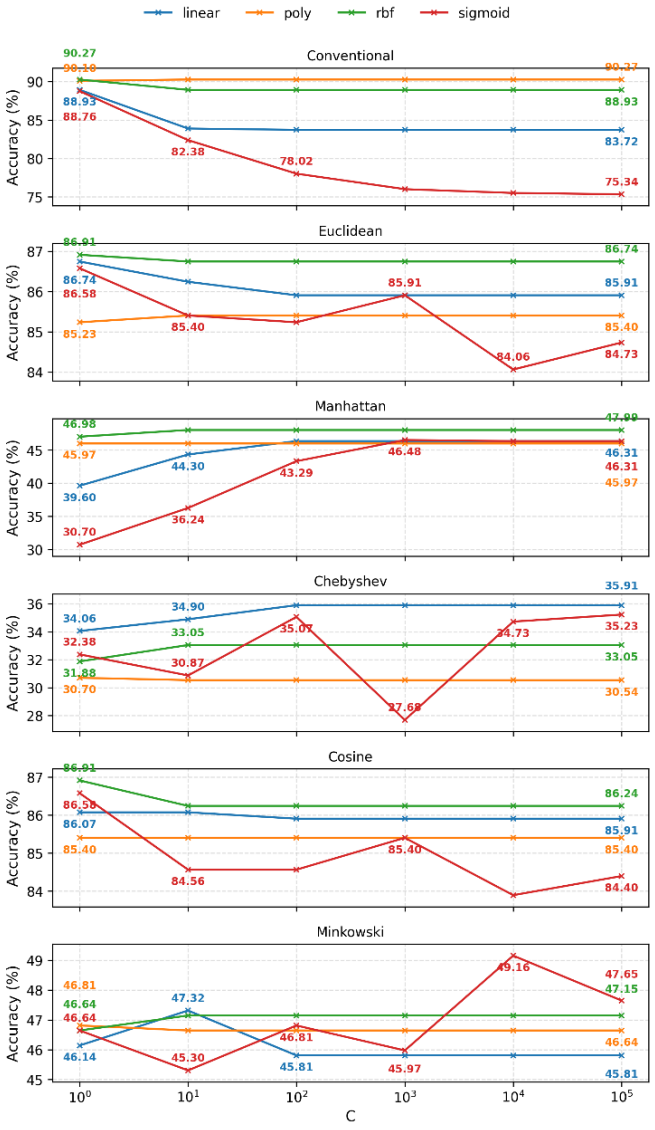

Fig. 9. Accuracy of the AG News dataset.

In addition to accuracy, the F1-score was also employed to account for class imbalance and provide a more comprehensive comparison of classification performance. Fig. 12 to Fig. 14 demonstrate the F1 score under an identical experimental configuration for all datasets. The exact numerical result is provided in the Appendix (see Table IV to Table VI).

Across all three datasets, the accuracy and the F1 score behaved very similarly under identical hyperparameter configurations within the same dataset. This indicates that the class imbalance did not distort the conclusion. The performance differences are not caused by the class imbalance, and confirms the robustness of the observation beyond accuracy alone.

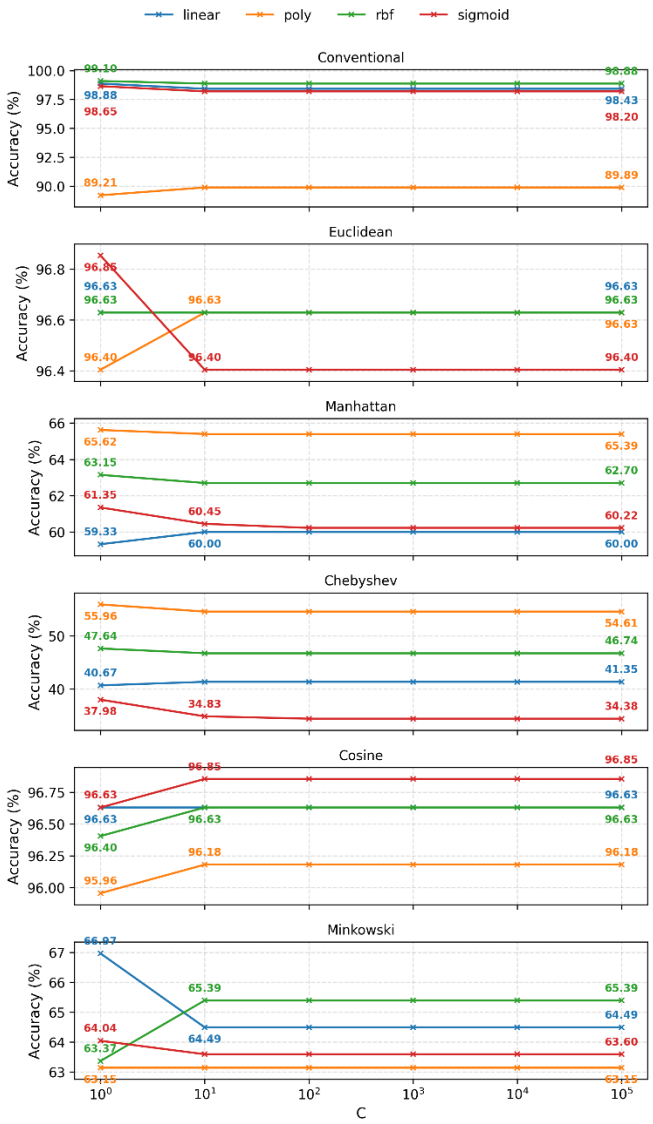

Fig. 10. Accuracy of the BBC News dataset.

Across all datasets, conventional SVM consistently achieves the peak in accuracy and in the F1 score, provided that the kernel and regularization parameters were carefully tuned. Specifically, the highest accuracy of 90.27% was obtained on the AG News dataset under the RBF kernel at C=1 and poly kernel at higher C values, 99.10% from BBC News by using the RBF kernel at C=1, and 95.48% on the DBpedia dataset under the linear kernel at C=10. However, the optimal kernel-regularization parameter combination varied across different datasets, showing its dataset-specific sensitivity.

Among the distance-based SVM variants, Euclidean and Cosine distance achieve competitive performance, closely approaching conventional SVM with only a little degradation in accuracy and F1 score across all datasets. Compared to the peak accuracy of conventional SVM, the performance

reduction remains bounded, with gaps of approximately 3-4% on the AG News dataset, 1-2% on the BBC News dataset, and 4-5% on the DBpedia dataset, indicating their strong discriminative capability. Their results remain consistently high, indicating their robustness in the text classification task.

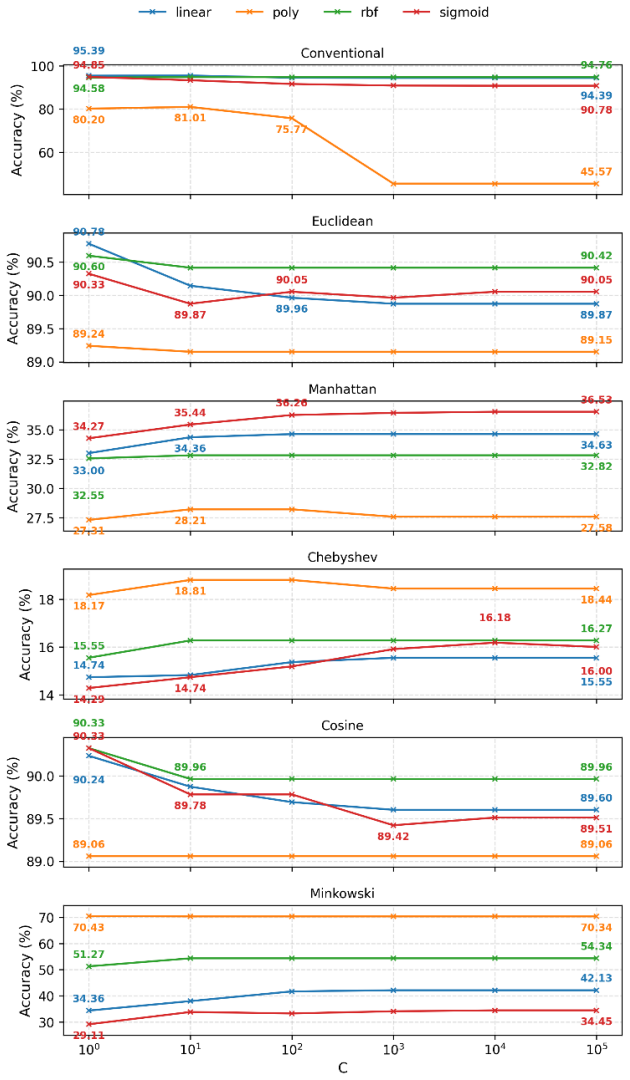

Fig. 11. Accuracy of the DBpedia Ontology classification dataset.

In contrast, the other distance-based SVM variants, including Manhattan, Chebyshev, and Minkowski distance-based SVM variants, consistently exhibit substantially lower performance across all datasets. These distances have low discriminative power in high-dimensional text representations, resulting in substantially lower accuracy and F1 score across kernel and regularization parameters.

Overall, the results indicate that the conventional SVM retains its position as the top performer when it is carefully tuned. The distance-based SVM, which uses Euclidean and Cosine distance, offers a strong trade-off between performance and robustness, making it suitable when excessive hyperparameter tuning is impractical and costly. This consistent trend was observed across all datasets with varying sizes and characteristics, indicating that the observed behaviors are not dataset-specific.

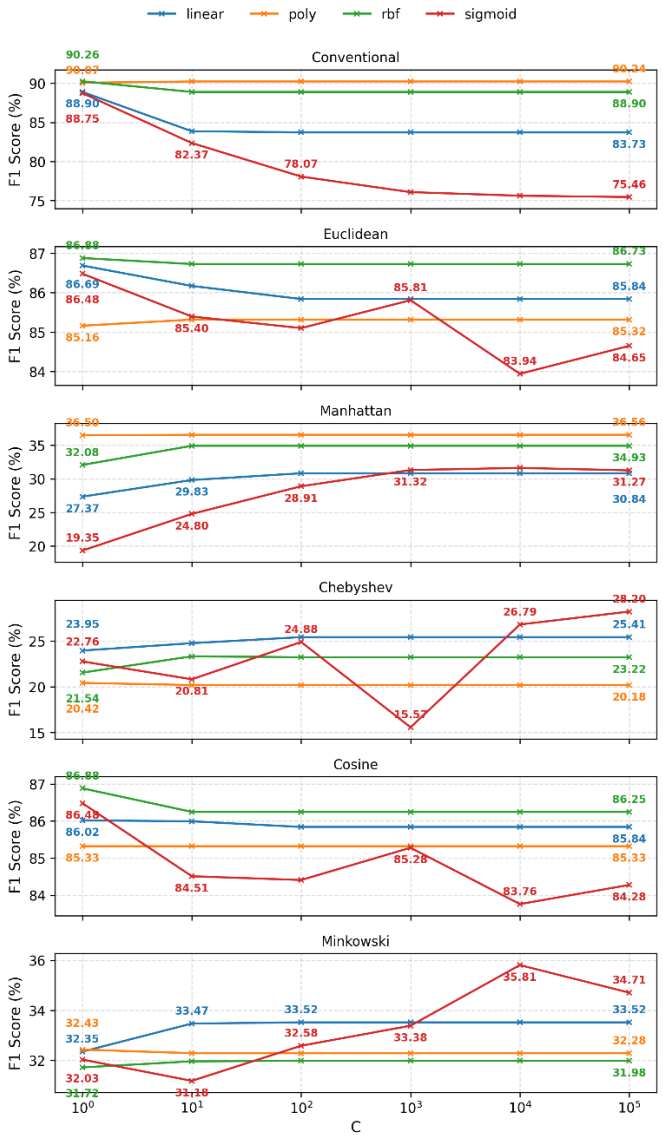

Fig. 12. F1 score of the AG News dataset.

B. Sensitivity of Conventional SVM to Kernel Functions and Regularization Parameters

This section evaluates the classification performance with respect to different kernel functions and regularization parameters, emphasizing the performance stability rather than the peak performance. The sensitivity of performance to hyperparameters is a critical issue for SVM classification tasks, as an inappropriate kernel function or regularization parameter may lead to substantial degradation in performance.

The conventional SVM exhibits a noticeable sensitivity to both kernel function and regularization parameter across all three datasets. While high accuracy and F1 score were obtained when the optimal combination of kernel and regularization parameter was chosen, the performance fluctuates significantly when the suboptimal combination of hyperparameters was

employed. For instance, on the AG News dataset, the sigmoid kernel demonstrates a noticeable performance degradation, dropping from 88.76% to 75.34% as the value of the regularization parameter increases. Similarly, for the BBC News dataset, the linear, RBF, and sigmoid kernels exhibit a near-optimal performance with accuracy approximately at 98-99%, but a clear performance gap is observed when switching the kernel to polynomial, where accuracy is maintained at 89-90%. In addition, on the DBpedia ontology classification dataset, the polynomial kernel demonstrates a severe decline in performance, dropping from 80.20% to 45.57% as the value of the regularization parameter increases. These observations highlight that conventional SVM performance was heavily dependent on dataset-specific hyperparameter tuning.

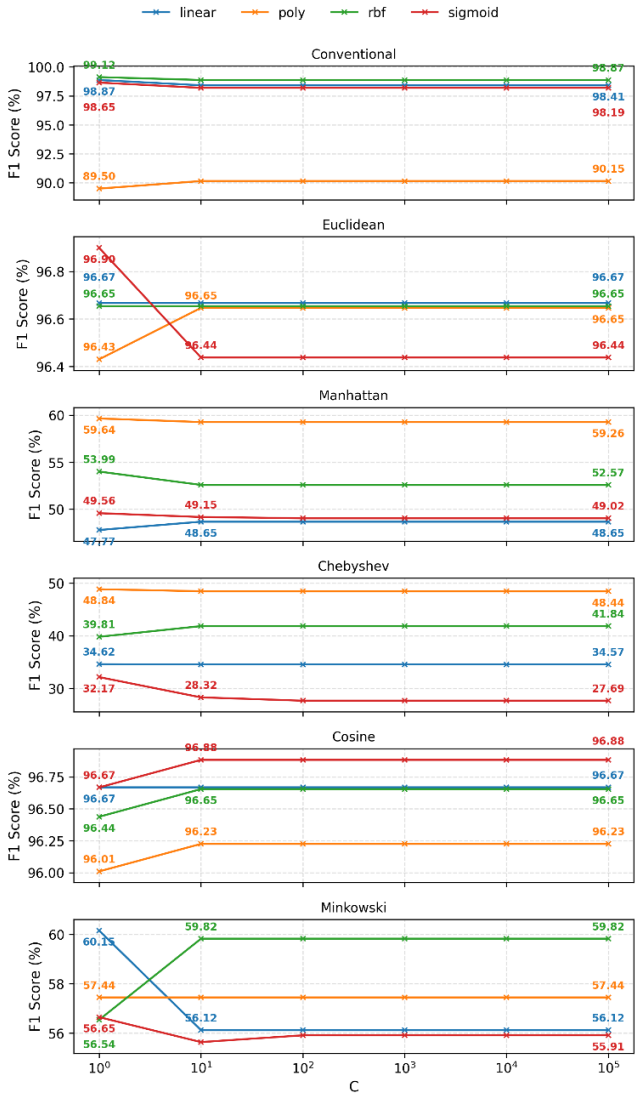

Fig. 13. F1 score of the BBC News dataset.

In contrast, the distance-based SVM variants, which utilize Euclidean and Cosine distance, demonstrate a substantially reduced fluctuation in performance across different kernels and regularization strengths. As illustrated in Fig. 9 to Fig. 14, their accuracy and F1 score exhibit nearly flat performance curves over all kernels and regularization strengths across all datasets. This minimal performance variation was bounded within a narrow margin, generally within 0-3%, suggesting that these methods exhibit more stable behavior under hyperparameter changing. This robustness is particularly evident in the DBpedia dataset, where Euclidean and Cosine exhibit a stable performance, even though the conventional SVM experiences a sharp decline in performance.

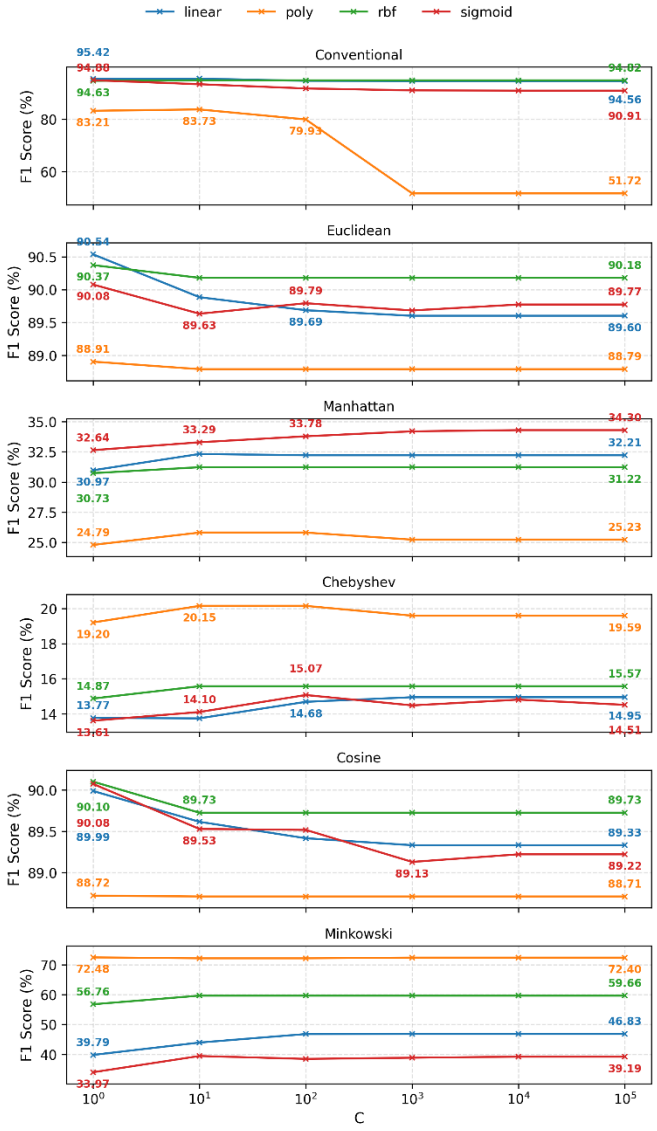

Fig. 14. F1 score of the DBpedia Ontology classification dataset.

The other distance-based SVM variants, which employed Manhattan, Chebyshev, and Minkowski distance, demonstrate low performance regardless of kernel and regularization parameter. The consistency across the different hyperparameter configurations suggests that for these distances, the choice of distance measures itself essentially limits the discriminative power. These distances may be less sensitive to the geometry in the high-dimensional, sparse TF-IDF vectors, making the average distance to support vectors not vary enough to support accurate classification. As a result, adjusting the kernel function and regularization parameter does not bring a significant impact on improving the classification performance.

Overall, these results clearly indicate the trade-off between peak performance and performance robustness. The conventional SVM can achieve peak performance when hyperparameter tuning is optimal. In contrast, Euclidean and Cosine-based SVM variants offer a more stable alternative, exhibiting lower variation across different kernel functions and regularization parameters, making them suitable for environments where exhaustive hyperparameter tuning was impractical.

C. Interpretation of Distance Metrics in High-Dimensional Text Classification

The text document was represented as a TF-IDF numerical representation. These representations produce high-dimensional and sparse vectors, where most of the values are zero. In this case, the choice of distance metric plays an important role in classification performance.

The Euclidean and Cosine distance-based SVM variants demonstrate strong and consistent performance across all datasets. Cosine distance measures the similarity of documents based on the orientation of feature vectors, emphasizing the relative term utilized pattern, making it effective in capturing the semantic similarity. This property plays an important role in text classification, as documents in the same category share a similar term distribution. Euclidean distance also performs well due to TF-IDF representation, which places the documents on a similar scale, making the distance comparison more meaningful. The document with similar patterns stays closer together, while the document from a different class is farther, leading to a reliable decision boundary for text classification.

Furthermore, distance-based SVMs may be more suitable for high-dimensional sparse data, given document representations such as TF-IDF, which primarily capture word-occurrence patterns rather than dense numerical relationships. In such feature spaces, the documents from the same classes often form local clusters with similar feature distributions. Although the performance is slightly lower than the peak accuracy of fine-tuned conventional SVM, their reduced sensitivity to kernel function and regularization parameters makes them more suitable in scenarios where extensive hyperparameter tuning or computational resources are limited.

In contrast, the distance-based SVM variants that employ Manhattan, Chebyshev, and Minkowski distance consistently perform weakly across all datasets. This behavior may relate to the geometric properties of high-dimensional sparse TF-IDF representations. Manhattan distance accumulates the absolute differences across all dimensions, making many small feature differences contribute equally to the final distance. Chebyshev distance only focuses on the feature with maximum distance, which means that the classification is dominated by one word, thus ignoring the overall document similarity, which significantly degrades the classification performance since the document's meaning comes from many words, not one. Similarly, the Minkowski distance with p=3 places greater emphasis on larger feature differences, which may distort the similarity relationships among sparse document vectors. As a result, these distances may be less effective in preserving meaningful document relationships in sparse text representations, regardless of kernel choice or regularization strength. This indicates that for distance-based SVM, the choice of distance plays a more important role than the kernel and regularization in high-dimensional classification datasets.

For the text classification task, if the environment is feasible for exhaustive hyperparameter tuning, the conventional SVM is recommended. However, when a rapid development is prioritized or when resources are limited and costly, the Euclidean and Cosine distance SVM variants are a preferable alternative. For example, it is suitable for the large-scale text system and resource-constrained environments.

D. Performance Comparison with LSTM

To further evaluate the effectiveness of the distance-based SVM variants, a comparison was conducted with a deep learning method, namely Long Short-Term Memory (LSTM).

Unlike the SVM-based methods, which were conducted under varied hyperparameter configurations, the LSTM model was trained using stochastic gradient-based optimization. As a result, the LSTM was reported with the accuracy and F1 Score under the same random seed as SVM, while the SVM-based methods were reported with their best performing configurations under varying hyperparameter scenarios. The results were reported in Fig. 15 and Fig. 16.

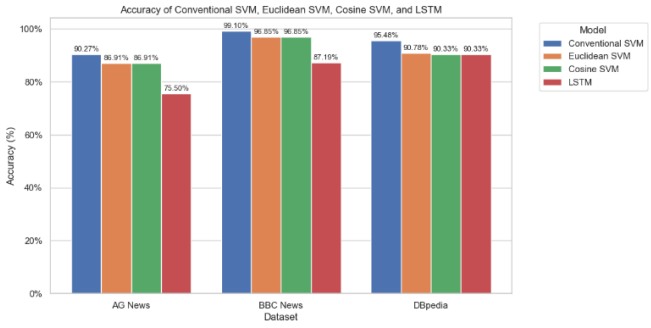

Fig. 15. Accuracy of Conventional SVM, Euclidean SVM, Cosine SVM, and LSTM.

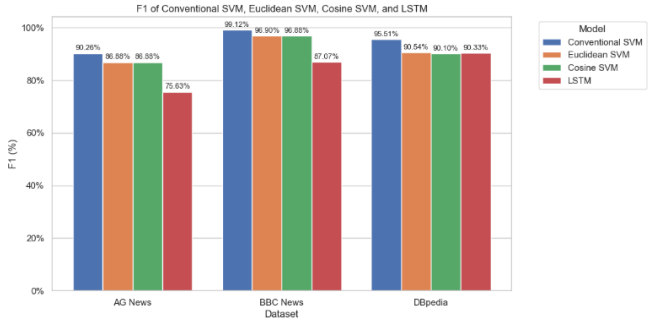

Fig. 16. F1 score of Conventional SVM, Euclidean SVM, Cosine SVM, and LSTM.

Across all the datasets, the SVM-based approaches consistently outperform LSTM. For the AG News dataset, the conventional SVM achieves an accuracy of 90.27%, and the F1 score of 90.26%; the Euclidean and Cosine-based SVMs achieve slightly lower but comparable performance with an accuracy of 86.91% and the F1 score of 86.88%. In contrast, the LSTM model achieves a notably lower accuracy of 75.50% and the F1 Score of 75.63%.

A similar trend was observed in the BBC News dataset. The conventional SVM achieves a near-perfect performance with an accuracy of 99.10% and the F1 score of 99.12%. The Euclidean and Cosine-based SVMs achieved a comparable performance with a slight degradation, with accuracy and F1 score falling between 96.8% and 97%. The LSTM model only achieves an accuracy of 87.19% and an F1 score of 87.07%, which is much lower than the SVM-based methods, indicating it is less effective in capturing the patterns compared to SVM-based methods.

For the DBpedia dataset, although the LSTM exhibits an obvious improvement compared to other datasets with an accuracy and F1 score of 90.33%, it still falls behind compared to SVM models with accuracy and F1 score above 95%, and comparable with the distance-based SVM, which includes Euclidean and Cosine that achieve similar accuracy and F1 Score.

Overall, the results demonstrate that the SVM-based models, including both conventional and distance-based variants, including Euclidean and Cosine, outperform the LSTM model under these three datasets. Notably, the Euclidean and Cosine-based SVM variants provide a strong balance between performance and stability, achieving results close to conventional SVM while reducing the sensitivity to hyperparameter changes.

IV. Conclusion

This study evaluates both conventional SVM and distance-based SVM, which incorporate various distance measures on three text classification datasets with varying size, class complexity, and text length. The aim is to assess both the peak performance and robustness to different kernel functions and regularization parameters.

From the experimental results observed across three text classification datasets, conventional SVM can achieve the highest peak accuracy provided the kernel functions and regularization parameters were fine-tuned. However, the performance exhibits substantial degradation when suboptimal hyperparameters are chosen. In contrast, the distance-based SVM variants that employed Euclidean and Cosine distance achieve a relatively slightly lower peak accuracy but maintain consistent performance across various kernels and regularization parameters. Other distance-based SVM variants, including Manhattan, Chebyshev, and Minkowski distance-based SVM, consistently performed poorly across all datasets.

The observed robustness of Euclidean and Cosine-based SVM variants reduces the sensitivity to kernel function and regularization parameter changes, making them suitable for scenarios where excessive hyperparameter tuning is computationally costly or impractical. This study provides practical guidance for selecting the SVM variants for text classification tasks, especially for resource-restricted environments.

In addition, the comparison with LSTM models shows that SVM-based models, including the conventional SVM and the high-performing distance-based variants (Euclidean and Cosine), consistently outperform LSTM models in both accuracy and F1 score across all datasets. These findings suggest that the traditional SVM-based models remain highly effective for the text classification task under the studied settings.

Future work can be extended by conducting a comprehensive evaluation with a deep learning approach, such as an advanced recurrent network or transformer-based models (e.g., BERT). The evaluation should be conducted under a unified experimental protocol, including consistent data preprocessing, train-test splits, and evaluation metrics to ensure a fair and systematic comparison across different models. In addition, further investigation can be carried out into other text representation techniques and large-scale text datasets to assess the generalization capability of the proposed approaches.

In short, the Euclidean- and Cosine-based SVM variants offer a robust and reliable alternative to conventional SVMs for text classification applications where hyperparameter tuning is limited.

References

[1] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” Lect. Notes Comput. Sci., vol. 1398, pp. 137–142, 1998.

[2] F. Sebastiani, “Machine Learning in Automated Text Categorization,” ACM Comput. Surv., vol. 34, no. 1, pp. 1–47, 2002.

[3] Y. Wahba, N. Madhavji, and J. Steinbacher, “A Comparison of SVM Against Pre-trained Language Models (PLMs) for Text Classification Tasks,” Lect. Notes Comput. Sci., vol. 13811 LNCS, pp. 304–313, 2023.

[4] A. Palanivinayagam, C. Z. El-Bayeh, and R. Damaševičius, “Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review,” Algorithms, vol. 16, no. 5, pp. 1–28, 2023.

[5] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” Phys. Rev. Lett., vol. 113, no. 3, pp. 1–5, 2014.

[6] L. H. Lee, C. H. Wan, R. Rajkumar, and D. Isa, “An enhanced Support Vector Machine classification framework by using Euclidean distance function for text document categorization,” Appl. Intell., vol. 37, no. 1, pp. 80–99, 2012.

[7] L. H. Lee, R. Rajkumar, L. H. Lo, C. H. Wan, and D. Isa, “Oil and gas pipeline failure prediction system using long range ultrasonic transducers and Euclidean-Support Vector Machines classification approach,” Expert Syst. Appl., vol. 40, no. 6, pp. 1925–1934, 2013.

[8] N. Amaya-Tejera, M. Gamarra, J. I. Vélez, and E. Zurek, “A distance-based kernel for classification via Support Vector Machines,” Front. Artif. Intell., vol. 7, 2024.

[9] W. Liu, S. Liang, and X. Qin, “Weighted p-norm distance t kernel SVM classification algorithm based on improved polarization,” Sci. Rep., vol. 12, no. 1, pp. 1–16, 2022.

[10] X. Zhang, J. Zhao, and Y. LeCun, “Character-level Convolutional Networks for Text Classification,” in Proceedings of the 29th International Conference on Neural Information Processing Systems, 2016, vol. 1, pp. 649–657.

[11] D. Greene and P. Cunningham, “Practical solutions to the problem of diagonal dominance in kernel document clustering,” ACM Int. Conf. Proceeding Ser., vol. 148, pp. 377–384, 2006.

APPENDIX

TABLE I. ACCURACY OF AG NEWS DATASET

Method	Kernel	C=1	C=10	C=100	C=1000	C=10000	C=100000
Conventional	linear	88.93%	83.89%	83.72%	83.72%	83.72%	83.72%
Conventional	poly	90.10%	90.27%	90.27%	90.27%	90.27%	90.27%
Conventional	rbf	90.27%	88.93%	88.93%	88.93%	88.93%	88.93%
Conventional	sigmoid	88.76%	82.38%	78.02%	76.01%	75.50%	75.34%
Euclidean	linear	86.74%	86.24%	85.91%	85.91%	85.91%	85.91%
Euclidean	poly	85.23%	85.40%	85.40%	85.40%	85.40%	85.40%
Euclidean	rbf	86.91%	86.74%	86.74%	86.74%	86.74%	86.74%
Euclidean	sigmoid	86.58%	85.40%	85.23%	85.91%	84.06%	84.73%
Manhattan	linear	39.60%	44.30%	46.31%	46.31%	46.31%	46.31%
Manhattan	poly	45.97%	45.97%	45.97%	45.97%	45.97%	45.97%
Manhattan	rbf	46.98%	47.99%	47.99%	47.99%	47.99%	47.99%
Manhattan	sigmoid	30.70%	36.24%	43.29%	46.48%	46.31%	46.31%
Chebyshev	linear	34.06%	34.90%	35.91%	35.91%	35.91%	35.91%
Chebyshev	poly	30.70%	30.54%	30.54%	30.54%	30.54%	30.54%
Chebyshev	rbf	31.88%	33.05%	33.05%	33.05%	33.05%	33.05%
Chebyshev	sigmoid	32.38%	30.87%	35.07%	27.68%	34.73%	35.23%
Cosine	linear	86.07%	86.07%	85.91%	85.91%	85.91%	85.91%
Cosine	poly	85.40%	85.40%	85.40%	85.40%	85.40%	85.40%
Cosine	rbf	86.91%	86.24%	86.24%	86.24%	86.24%	86.24%
Cosine	sigmoid	86.58%	84.56%	84.56%	85.40%	83.89%	84.40%
Minkowski	linear	46.14%	47.32%	45.81%	45.81%	45.81%	45.81%
Minkowski	poly	46.81%	46.64%	46.64%	46.64%	46.64%	46.64%
Minkowski	rbf	46.64%	47.15%	47.15%	47.15%	47.15%	47.15%
Minkowski	sigmoid	46.64%	45.30%	46.81%	45.97%	49.16%	47.65%

TABLE II. ACCURACY OF BBC NEWS DATASET

Method	Kernel	C=1	C=10	C=100	C=1000	C=10000	C=100000
Conventional	linear	98.88%	98.43%	98.43%	98.43%	98.43%	98.43%
Conventional	poly	89.21%	89.89%	89.89%	89.89%	89.89%	89.89%
Conventional	rbf	99.10%	98.88%	98.88%	98.88%	98.88%	98.88%
Conventional	sigmoid	98.65%	98.20%	98.20%	98.20%	98.20%	98.20%
Euclidean	linear	96.63%	96.63%	96.63%	96.63%	96.63%	96.63%
Euclidean	poly	96.40%	96.63%	96.63%	96.63%	96.63%	96.63%
Euclidean	rbf	96.63%	96.63%	96.63%	96.63%	96.63%	96.63%
Euclidean	sigmoid	96.85%	96.40%	96.40%	96.40%	96.40%	96.40%
Manhattan	linear	59.33%	60.00%	60.00%	60.00%	60.00%	60.00%
Manhattan	poly	65.62%	65.39%	65.39%	65.39%	65.39%	65.39%
Manhattan	rbf	63.15%	62.70%	62.70%	62.70%	62.70%	62.70%
Manhattan	sigmoid	61.35%	60.45%	60.22%	60.22%	60.22%	60.22%

Chebyshev	linear	40.67%	41.35%	41.35%	41.35%	41.35%	41.35%
Chebyshev	poly	55.96%	54.61%	54.61%	54.61%	54.61%	54.61%
Chebyshev	rbf	47.64%	46.74%	46.74%	46.74%	46.74%	46.74%
Chebyshev	sigmoid	37.98%	34.83%	34.38%	34.38%	34.38%	34.38%
Cosine	linear	96.63%	96.63%	96.63%	96.63%	96.63%	96.63%
Cosine	poly	95.96%	96.18%	96.18%	96.18%	96.18%	96.18%
Cosine	rbf	96.40%	96.63%	96.63%	96.63%	96.63%	96.63%
Cosine	sigmoid	96.63%	96.85%	96.85%	96.85%	96.85%	96.85%
Minkowski	linear	66.97%	64.49%	64.49%	64.49%	64.49%	64.49%
Minkowski	poly	63.15%	63.15%	63.15%	63.15%	63.15%	63.15%
Minkowski	rbf	63.37%	65.39%	65.39%	65.39%	65.39%	65.39%
Minkowski	sigmoid	64.04%	63.60%	63.60%	63.60%	63.60%	63.60%

TABLE III. ACCURACY OF DBPEDIA ONTOLOGY CLASSIFICATION DATASET

Method	Kernel	C=1	C=10	C=100	C=1000	C=10000	C=100000
Conventional	linear	95.39%	95.48%	94.48%	94.39%	94.39%	94.39%
Conventional	poly	80.20%	81.01%	75.77%	45.57%	45.57%	45.57%
Conventional	rbf	94.58%	94.76%	94.76%	94.76%	94.76%	94.76%
Conventional	sigmoid	94.85%	93.31%	91.59%	90.87%	90.78%	90.78%
Euclidean	linear	90.78%	90.14%	89.96%	89.87%	89.87%	89.87%
Euclidean	poly	89.24%	89.15%	89.15%	89.15%	89.15%	89.15%
Euclidean	rbf	90.60%	90.42%	90.42%	90.42%	90.42%	90.42%
Euclidean	sigmoid	90.33%	89.87%	90.05%	89.96%	90.05%	90.05%
Manhattan	linear	33.00%	34.36%	34.63%	34.63%	34.63%	34.63%
Manhattan	poly	27.31%	28.21%	28.21%	27.58%	27.58%	27.58%
Manhattan	rbf	32.55%	32.82%	32.82%	32.82%	32.82%	32.82%
Manhattan	sigmoid	34.27%	35.44%	36.26%	36.44%	36.53%	36.53%
Chebyshev	linear	14.74%	14.83%	15.37%	15.55%	15.55%	15.55%
Chebyshev	poly	18.17%	18.81%	18.81%	18.44%	18.44%	18.44%
Chebyshev	rbf	15.55%	16.27%	16.27%	16.27%	16.27%	16.27%
Chebyshev	sigmoid	14.29%	14.74%	15.19%	15.91%	16.18%	16.00%
Cosine	linear	90.24%	89.87%	89.69%	89.60%	89.60%	89.60%
Cosine	poly	89.06%	89.06%	89.06%	89.06%	89.06%	89.06%
Cosine	rbf	90.33%	89.96%	89.96%	89.96%	89.96%	89.96%
Cosine	sigmoid	90.33%	89.78%	89.78%	89.42%	89.51%	89.51%
Minkowski	linear	34.36%	37.97%	41.68%	42.13%	42.13%	42.13%
Minkowski	poly	70.43%	70.34%	70.34%	70.34%	70.34%	70.34%
Minkowski	rbf	51.27%	54.34%	54.34%	54.34%	54.34%	54.34%
Minkowski	sigmoid	29.11%	33.82%	33.27%	34.09%	34.45%	34.45%

TABLE IV. F1 SCORE OF AG NEWS DATASET

Method	Kernel	C=1	C=10	C=100	C=1000	C=10000	C=100000
Conventional	linear	88.90%	83.88%	83.73%	83.73%	83.73%	83.73%
Conventional	poly	90.07%	90.24%	90.24%	90.24%	90.24%	90.24%
Conventional	rbf	90.26%	88.90%	88.90%	88.90%	88.90%	88.90%
Conventional	sigmoid	88.75%	82.37%	78.07%	76.08%	75.63%	75.46%
Euclidean	linear	86.69%	86.17%	85.84%	85.84%	85.84%	85.84%
Euclidean	poly	85.16%	85.32%	85.32%	85.32%	85.32%	85.32%
Euclidean	rbf	86.88%	86.73%	86.73%	86.73%	86.73%	86.73%
Euclidean	sigmoid	86.48%	85.40%	85.10%	85.81%	83.94%	84.65%
Manhattan	linear	27.37%	29.83%	30.84%	30.84%	30.84%	30.84%
Manhattan	poly	36.50%	36.56%	36.56%	36.56%	36.56%	36.56%
Manhattan	rbf	32.08%	34.93%	34.93%	34.93%	34.93%	34.93%
Manhattan	sigmoid	19.35%	24.80%	28.91%	31.32%	31.64%	31.27%
Chebyshev	linear	23.95%	24.75%	25.41%	25.41%	25.41%	25.41%
Chebyshev	poly	20.42%	20.18%	20.18%	20.18%	20.18%	20.18%
Chebyshev	rbf	21.54%	23.32%	23.22%	23.22%	23.22%	23.22%
Chebyshev	sigmoid	22.76%	20.81%	24.88%	15.57%	26.79%	28.20%
Cosine	linear	86.02%	85.99%	85.84%	85.84%	85.84%	85.84%
Cosine	poly	85.33%	85.33%	85.33%	85.33%	85.33%	85.33%
Cosine	rbf	86.88%	86.25%	86.25%	86.25%	86.25%	86.25%
Cosine	sigmoid	86.48%	84.51%	84.41%	85.28%	83.76%	84.28%
Minkowski	linear	32.35%	33.47%	33.52%	33.52%	33.52%	33.52%
Minkowski	poly	32.43%	32.28%	32.28%	32.28%	32.28%	32.28%
Minkowski	rbf	31.72%	31.96%	31.98%	31.98%	31.98%	31.98%
Minkowski	sigmoid	32.03%	31.18%	32.58%	33.38%	35.81%	34.71%

TABLE V. F1 SCORE OF BBC NEWS DATASET

Method	Kernel	C=1	C=10	C=100	C=1000	C=10000	C=100000
Conventional	linear	98.87%	98.41%	98.41%	98.41%	98.41%	98.41%
Conventional	poly	89.50%	90.15%	90.15%	90.15%	90.15%	90.15%
Conventional	rbf	99.12%	98.87%	98.87%	98.87%	98.87%	98.87%
Conventional	sigmoid	98.65%	98.19%	98.19%	98.19%	98.19%	98.19%
Euclidean	linear	96.67%	96.67%	96.67%	96.67%	96.67%	96.67%
Euclidean	poly	96.43%	96.65%	96.65%	96.65%	96.65%	96.65%
Euclidean	rbf	96.65%	96.65%	96.65%	96.65%	96.65%	96.65%
Euclidean	sigmoid	96.90%	96.44%	96.44%	96.44%	96.44%	96.44%
Manhattan	linear	47.77%	48.65%	48.65%	48.65%	48.65%	48.65%
Manhattan	poly	59.64%	59.26%	59.26%	59.26%	59.26%	59.26%
Manhattan	rbf	53.99%	52.57%	52.57%	52.57%	52.57%	52.57%
Manhattan	sigmoid	49.56%	49.15%	49.02%	49.02%	49.02%	49.02%

Chebyshev	linear	34.62%	34.57%	34.57%	34.57%	34.57%	34.57%
Chebyshev	poly	48.84%	48.44%	48.44%	48.44%	48.44%	48.44%
Chebyshev	rbf	39.81%	41.84%	41.84%	41.84%	41.84%	41.84%
Chebyshev	sigmoid	32.17%	28.32%	27.69%	27.69%	27.69%	27.69%
Cosine	linear	96.67%	96.67%	96.67%	96.67%	96.67%	96.67%
Cosine	poly	96.01%	96.23%	96.23%	96.23%	96.23%	96.23%
Cosine	rbf	96.44%	96.65%	96.65%	96.65%	96.65%	96.65%
Cosine	sigmoid	96.67%	96.88%	96.88%	96.88%	96.88%	96.88%
Minkowski	linear	60.15%	56.12%	56.12%	56.12%	56.12%	56.12%
Minkowski	poly	57.44%	57.44%	57.44%	57.44%	57.44%	57.44%
Minkowski	rbf	56.54%	59.82%	59.82%	59.82%	59.82%	59.82%
Minkowski	sigmoid	56.65%	55.63%	55.91%	55.91%	55.91%	55.91%

TABLE VI. F1 SCORE OF DBPEDIA ONTOLOGY CLASSIFICATION DATASET

Method	**Kernel**	**C=1**	**C=10**	**C=100**	**C=1000**	**C=10000**	**C=100000**
Conventional	linear	95.42%	95.51%	94.63%	94.56%	94.56%	94.56%
Original	poly	83.21%	83.73%	79.93%	51.72%	51.72%	51.72%
Original	rbf	94.63%	94.82%	94.82%	94.82%	94.82%	94.82%
Original	sigmoid	94.88%	93.35%	91.73%	91.01%	90.91%	90.91%
Euclidean	linear	90.54%	89.89%	89.69%	89.60%	89.60%	89.60%
Euclidean	poly	88.91%	88.79%	88.79%	88.79%	88.79%	88.79%
Euclidean	rbf	90.37%	90.18%	90.18%	90.18%	90.18%	90.18%
Euclidean	sigmoid	90.08%	89.63%	89.79%	89.68%	89.77%	89.77%
Manhattan	linear	30.97%	32.32%	32.21%	32.21%	32.21%	32.21%
Manhattan	poly	24.79%	25.81%	25.81%	25.23%	25.23%	25.23%
Manhattan	rbf	30.73%	31.22%	31.22%	31.22%	31.22%	31.22%
Manhattan	sigmoid	32.64%	33.29%	33.78%	34.19%	34.30%	34.30%
Chebyshev	linear	13.77%	13.74%	14.68%	14.95%	14.95%	14.95%
Chebyshev	poly	19.20%	20.15%	20.15%	19.59%	19.59%	19.59%
Chebyshev	rbf	14.87%	15.57%	15.57%	15.57%	15.57%	15.57%
Chebyshev	sigmoid	13.61%	14.10%	15.07%	14.48%	14.81%	14.51%
Cosine	linear	89.99%	89.62%	89.42%	89.33%	89.33%	89.33%
Cosine	poly	88.72%	88.71%	88.71%	88.71%	88.71%	88.71%
Cosine	rbf	90.10%	89.73%	89.73%	89.73%	89.73%	89.73%
Cosine	sigmoid	90.08%	89.53%	89.52%	89.13%	89.22%	89.22%
Minkowski	linear	39.79%	43.93%	46.79%	46.83%	46.83%	46.83%
Minkowski	poly	72.48%	72.20%	72.20%	72.40%	72.40%	72.40%
Minkowski	rbf	56.76%	59.66%	59.66%	59.66%	59.66%	59.66%
Minkowski	sigmoid	33.97%	39.42%	38.45%	38.84%	39.19%	39.19%