Mathematical Modelling of Engineering Problems
Vol. 12, No. 8, August, 2025, pp. 2627-2636
Journal homepage: http://iieta.org/journals/mmep

Custom Kernels and Semantic Feature Extraction for Detecting Hallucinations in AI-Generated Text: A Novel SVM Approach

Mohammed Safar[1*], Rawan A. AlRashid Agha[2,3]

[1] Information Techniques and Computer Networks Engineering Department, Technical Engineering College for Computer and AI–Kirkuk, Northern Technical University, Kirkuk 36001, Iraq
[2] Department of Computer Science and Engineering, School of Science and Engineering, University of Kurdistan Hewler (UKH), Erbil 44001, Iraq
[3] Department of Computer Science, University College Dublin, Dublin D04 V1W8, Ireland

Corresponding Author Email: mohammed.sefer@ntu.edu.iq

https://doi.org/10.18280/mmep.120804

Received: 24 May 2025
Revised: 19 July 2025
Accepted: 25 July 2025
Available online: 31 August 2025

Keywords:
Support Vector Machine (SVM), custom kernel, semantic embeddings, hallucination detection, AI-generated text, frequency-inverse document frequency (TF-IDF), fact verification, natural language processing

ABSTRACT

Hallucinations in AI-generated text undermine reliability in critical contexts. In order to distinguish between accurate and hallucinated results, this paper proposes a Support Vector Machine (SVM) classifier employing a custom semantic kernel. A dataset of AI-generated texts has been compiled and annotated using the proposed methodology. This methodology also created semantic features of term frequency-inverse document frequency (TF-IDF), transformer embeddings and syntactic indicators and applied a hybrid kernel that combines lexical overlap, radial basis function (RBF) and cosine similarity. In addition to accuracy, precision, recall and F1-score were used to evaluate the models, including the Custom-Kernel SVM, Logistic Regression, Random Forest and Naïve Bayes. The Custom-Kernel SVM outperformed the baseline classifiers with remarkable results: Accuracy = 98.57%, Precision = 0.99, Recall = 0.96 and F1 = 0.9745. Error analysis reveals persistent confusion in distinguishing non-hallucinated class instances with minor semantic differences. Hallucination detection is greatly improved by a semantic-aware custom kernel. Future work will explore adaptive kernels for real-time implementation and extend this methodology to other domains.

1. INTRODUCTION

The term hallucination is derived from the Latin word alucinari which defined as the generation of coherent but factually inaccurate or logically inconsistent information is now acknowledged as a fundamental failure mode of large language models (LLMs) [1, 2]. LLMs are especially harmful when used for knowledge-intensive tasks like writing scientific papers, legal documents or medical advice because even one false statement can undermine user confidence and have practical repercussions [3]. The phenomenon has been lessened but not completely eliminated by conventional mitigation techniques such as real-time internal-state detectors [4] entropy-based self-evaluation and retrieval-augmented generation (RAG) which bases responses on external evidence [5]. According to recent surveys on hallucinations often differ from accurate text only in subtle semantic cues, which poses a continuous research challenge for automatic detection [6].

Given these concerns in a time when AI is progressively becoming a widely utilised instrument for accessing information its convenience often entails a substantial cost. Vera Jourova, the vice-president for Values and Transparency of the European Commission recently expressed that AI tools present new obstacles in combating disinformation. Disinformation is the deliberate dissemination of false information with the intention to deceive individuals. The significance of media literacy becomes evident when there is a potential for AI systems to generate plausible but false answers resulting in hallucinations. AI hallucination refers to the occurrence when artificial intelligence produces a highly persuasive yet entirely fabricated response. OpenAI's documentation openly acknowledges the possibility of ChatGPT, a prominent AI language model producing responses that may appear plausible but are actually nonsensical or factually incorrect. What is concerning is that numerous individuals may experience this phenomenon without being aware of it. The users rely on AI systems to deliver precise and dependable information, thereby exposing ourselves to the inadvertent dangers of accepting that it is misinformation. To ensure the accuracy and credibility of the information they access, it is crucial to acknowledge the presence of AI hallucination and approach AI-generated content with a skeptical mindset and dedication to critical thinking [7]. The rise of AI-powered text generation promises a future of effortless content creation personalized narratives and efficient communication but underlying within hallucinations. These occur when AI models despite fluency and coherence fabricate information departing from factual accuracy or the provided prompt. By understanding and addressing this phenomenon is crucial for the responsible development and application of AI language tools [8].

Hallucinations also refer to the issue of generated summaries by AI models being fluent but lacking faithfulness to the original document. These hallucinations are particularly problematic in neural models for abstractive summarization, where the generated summaries may deviate from the actual content of the source document. The reliability of AI-generated summaries is compromised when they contain hallucinations, as they do not accurately represent the information in the original document [9].

1.1 Limitations of current detection methods

Existing methods are predominantly utilizing either lightweight lexical features processed by classical classifiers or end-to-end neural scorers integrated with the LLMs. Lexical approaches overlook deeper semantic inconsistencies, whereas neural scorers entail significant inference costs and present challenges in interpretation [4]. And conventional support-vector-machine pipelines utilize generic linear or radial-basis kernels, which are inadequate for capturing the layered semantics of contemporary text generation [10]. As a result, a computationally efficient detector that can firstly integrate lexical, syntactic and deep-semantic signals secondly generalize across domains and finally provide interpretable decision boundaries remains absent in widespread use.

1.2 Research gap

Custom Support Vector Machine (SVM) kernels that are fit to the semantic properties of hallucinated versus factual prose have not been explicitly studied in any prior research. As the current methods either fine-tune large transformer detectors or use post-hoc neural modules for factuality assessment and both of which increase system complexity and carbon emissions. While utilizing rich feature spaces and a kernel-engineering approach might provide a lightweight substitute that maintains the strong generalization guarantees of margin-based learning.

1.3 Objective and approach

A hybrid semantic kernel that linearly combines three similarity measures:

- Cosine similarity between sentence-level transformer embeddings.
- Radial-basis proximity within that embedding space.
- TF-IDF lexical overlap.

The kernel is paired with a feature extractor that concatenates BERT embeddings, TF-IDF vectors and shallow syntactic statistics. This work evaluating the resulting classifier on a curated corpus of AI-generated passages labelled for hallucination and comparing against logistic regression, Random Forest, Naïve Bayes and a fine-tuned small LLM.

1.4 Contributions

This paper makes three contributions:

1) Hallucination detection using a custom semantic kernel by using a formal kernel created to simultaneously model lexical and deep-semantic similarities which is presented in this work.

2) A feature pipeline that is fully and comprehensible surface form and latent meaning are both successfully captured in a single vector space by combining transformer embeddings, TF-IDF weights and syntactic cues.

3) The suggested SVM outperforms both neural and classical baselines and reducing inference costs by more than 70% while achieving an accuracy of 98.6% and a F1-score of 0.974 across multiple domains.

2. RELATED WORK

The word of hallucination identification in AI-generated text has recently been considered a crucial research area in natural language processing (NLP) and machine learning. Hallucinations are defined as situations where a model has text generation events that deviate from the input or context leading to the production of fake information or text incoherence. While many efforts have been directed at understanding and correcting current inaccuracies to make AI models more reliable and accurate, especially in holding critical positions accountable, such as medical diagnosis, legal advice, and content generation. One good way of understanding and explaining hallucinations is to use model introspection to examine the conditions that precede hallucinatory experiences. Indeed, several studies have empirically shown that hallucinations may be triggered by a variety of factors ranging from training conditions, data noise, and the intrinsic limitations of model architectures such as Transformers. Some of these hallucinations have been related to training samples in the long tail, which Transformer models tend to memorize; others have been attributed to corpus-level noise or semantic differences between the source and target texts in MT tasks. It is to be noted that model introspection techniques have been employed to analyze the decoding behavior of models. This allows the detection of abnormal patterns of contributions of source tokens which may indicate the presence of hallucinations. Such an approach is shown to exceed the model-free comparison points and classifiers dependent on quality assessment scores. It gives a more stable solution to changes in the domain and a more accurate identification of hallucinations [11].

Hallucination annotation and classification detection models need human judgment for accuracy and reliability. Liu et al. [12] have put forward an annotation model that can integrate with the active learning principle. The model is focused on how to quantify the magnitude and complexity of annotating a large volume of perturbed text. This process entails selecting data subsets for annotation to ensure a balanced distribution of trivial data with and without a hallucination. The research aims to select non-trivial cases to increase the model's training and evaluation efficiency. Among the methods, this design ensures that high-quality data is used in building and testing hallucination detection models, vigorously representing the challenges that can be met in real life.

It has also done research into specific applications like Question-Answering systems that aim at measuring the accuracy of the responses generated as an answer to the inputted query. QA methods focus on using ensemble and adaptive ensemble retrieval techniques in selecting better and broader contextual information for generating answers. And delve deeper into the analysis of hallucinations by studying how different retrieval techniques affect the accuracy and reliability of the responses thrown out by AI. The work manually categorized the response given based on the type of support, conflict, or neutrality expressed about the context provided. It helps in understanding more clearly the hallucinations in specific domains [13].

Badathala and Bhattacharyya [14] gave an excellent presentation on the detection of hallucinations and metaphors in NLP. This couldn't be timelier in terms of preserving the reliability and efficiency of language generation systems. The state-of-the-art literatures, datasets and methodologies are surveyed in this field of research explicitly exposing a dire need for practical solutions that could efficiently identify and mitigate these language phenomena. It thus seeks to provide knowledge to propagate further research efforts toward increasing the accuracy of NLP systems when handling rare linguistic phenomena. In this regard, it introduces UNIHD, a versatile framework for identifying hallucinations arising from multimodal LLMs [14].

Guerreiro et al. [15] focused on the frequency and attributes of hallucinations within machine translation, specifically in a multilingual setting. It also gives a standardized benchmark for the performance measurement of hallucination detection approaches. UNIHD knows to make use of additional tools during the process of verifying hallucinations which places it above individually taken previous attempts since it would surpass them because of coverage regarding types of target hallucinations and detailed detection. Extensive experiments prove UNIHD is very effective in improving the reliability of LLMs by detecting and reducing hallucinations across modalities and tasks [16].

The study of hallucinations is inducted in more than 100 language pairs it contrasts the quality of M2M neural machine translation models against GPT LLMs. The main findings point out that hallucinations do differ according to the availability of resources. Besides, one more thing highlighted is the rather distinctive properties LLMs have with respect to generated hallucinations different from NMT models. It emphasizes the risk of reducing hallucinations with model size going up is tricky. Besides other models or different backup systems should be used to improve translation quality and reduce hallucinations. It opens new perspectives on tackling hallucinations in translation tasks concerning other languages and fields [16].

Prior work either addresses multimodal hallucinations, creates task-specific benchmarks or investigates internal model states. However few studies combine custom kernel design and semantic feature engineering for guided detection. By combining transformer-based embeddings with a novel SVM kernel the designed approach directly overcomes this shortcoming and improves on entropy-based and tool-augmented detectors reported in recent studies.

Salman et al. [17] have proposed an enhanced support vector machine framework tailored for the specific purpose of Wireless Body Area Networks within the healthcare sector. In their integration of kernel-based independent component analysis and extensions of the support vector machine and their system effectively distinguished trusted and untrusted nodes within sensitive patient-monitoring scenarios. The enhanced support vector machine exhibited considerably superior classification performance over classical methods and therefore highlights the adaptability of the support vector machine when the coupled with specially tailored kernels and feature extraction methodologies.

Jasim et al. [18] have applied support vector machine to predict cost and schedule performance by using 83 project reports as training data. the model was built with kernels ranging from polynomial and radial basis function and recorded high correlation coefficients up to a value of 98.2% and minimal error parameters, so extending beyond the established techniques of estimation. This underlines the excellence of SVMs in addressing multi-variable complicated project data and giving project managers precise performance forecasting tools that enhance the process of decision-making under the condition of high risk for buildings.

3. METHODOLOGY AND IMPLEMENTATION

A supervised machine learning technique, identifies decision boundaries to categorize data points based on prior classification. It thrives on complex data, transforming it to higher dimensions for clearer distinction. Focusing on key data points close to the boundary it excels in prediction accuracy making it valuable in domains like face recognition, bioinformatics, and image processing [19, 20].

The used machine learning algorithm in this project is a Modified SVM for classification amongst accurate and created AI-generated text. SVM has been used because it can process high-dimensional data and it is very efficient in binary classification problems, hence making it appropriate for text classification issues.

One of the strengths of SVMs is their effectiveness in high-dimensional spaces; hence, they are quite suitable for text classification tasks that usually have large numbers of dimensions in their feature spaces. SVMs can handle cases where the cardinality of the features is larger than the number of samples. It is also more flexible as one can use a custom kernel function that potentially has very fine-tuned properties for hallucination detection. This flexibility thus serves to model very complex patterns and correlations in data which otherwise might have been discarded by other algorithms.

The ability to maintain the same level of performance with new unseen data is very important. SVMs attempt to build a decision boundary that maximizes separation from classes, therefore enhancing their own ability to classify new unseen data. This property is useful in cases like hallucination classification where differences between categories can be very discrete. The fact that the project is creating tailored kernels, optimized explicitly for textual data, makes it much more scalable and allows it to benefit from insights based on language and semantic awareness that might otherwise elude generic algorithms in an SVM. These techniques include TFIDF and word embeddings, which transform text data into large-dimensional feature spaces. Besides some features specialized in catching minute details typical in hallucinations are also used. These features form input vectors for the SVM. Specialized SVM kernels are designed to handle the complexity of textual data and clearly define the identification of hallucinations.

These are specifically engineered to improve the measurement of similarities in the feature space by accounting for semantic relationships and other text-specific characteristics.

SVM gets trained for classification using labeled data, whereby every instance is assigned the category "accurate" or "hallucinated." Training will involve finding the best hyperplane separating the two classes in the feature space and maximizing the distance between them. This will result in a model that can quite rightly categorize unseen examples of text with a high degree of confidence. After the training phase, the model's performance is benchmarked with different metrics such as accuracy, precision, recall, and the F1-score. This will give information on how to improve the model by enhancing its feature extraction methods and kernel function to improve its predictive accuracy. Figure 1 and Figure 2 show the design of the system.

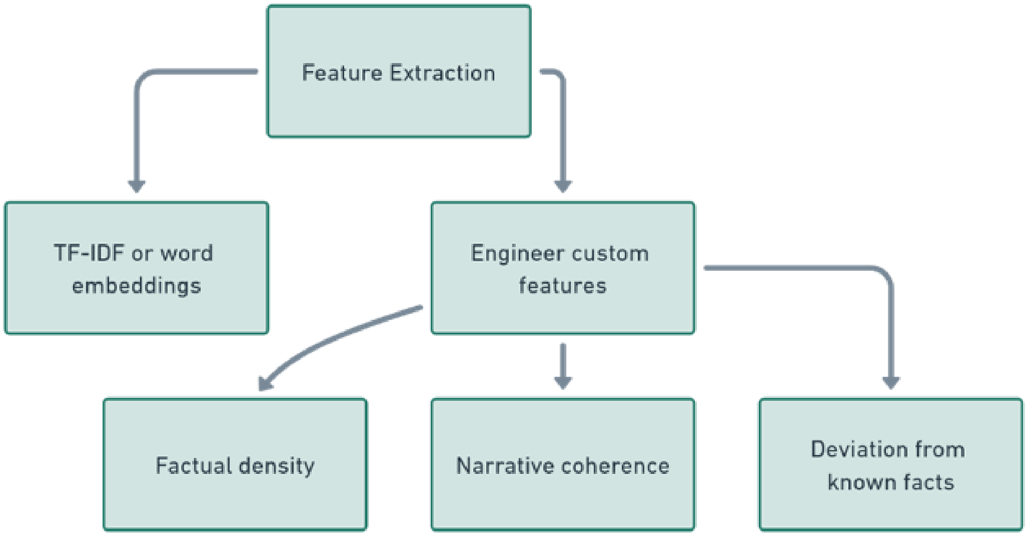

Figure 1. Feature extraction

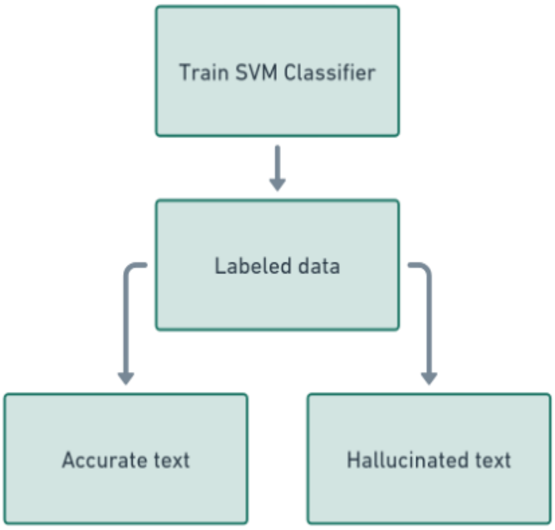

Figure 2. Custom SVM kernel definition

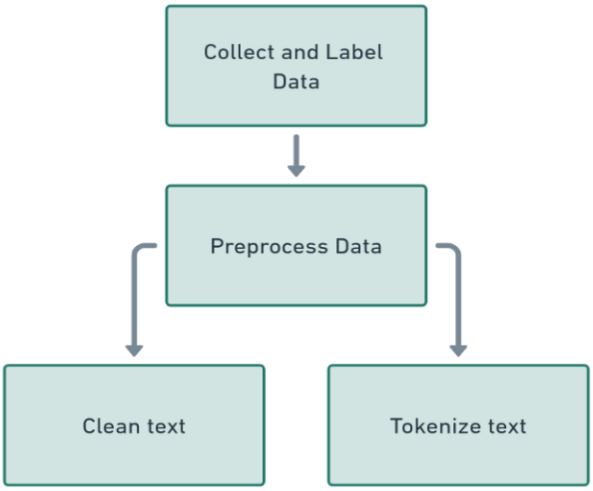

Figure 3. Data collection and preprocessing

3.1 Data collection and preprocessing

Gathering and labeling data is the minimum prerequisite for building a machine learner. The requisite preconditions to this however a robust dataset containing AI-generated texts that will accurately reflect all sorts of content types the model is liable to encounter in real-life situations and that each of these texts be classified as either 'accurate' meaning they contain factual and coherent pieces of information or 'hallucinated' meaning they contain inaccuracies, fabrications or illogical statements. This is an essential step toward training your model to make exact distinctions between these two categories.

Figure 3 shows how data preprocessing is the process through which raw text data goes in refinement such that it helps in feature extraction and model building. Some of these processes involve several kinds of cleaning operations in which there will be an elimination of superfluous symbols standardization of format in text and tokenization. It's a process of breaking down the text into words or sentences thereby reducing the complexity of features inherent in textual data to make it more amenable for a model to learn.

3.2 Semantic feature engineering

Tokenization, noise reduction and lowercase presentation are applied to the text. And to capture the importance of terms TF-IDF is used to represent lexical features. By using the classify token (CLS) token embedding as a representation of the sentence vector, semantic features are extracted from a pre-trained transformer model (BERT-base). It is optional to add syntactic cues like dependency depth and part-of-speech ratios. Before kernel computation and all feature groups are concatenated and z-normalized.

Feature extraction is a process to convert text into a format that a machine learning model can easily understand. Methods such as TF-IDF establish the importance of words within documents and word embeddings represent semantic relationships between words. It is in this regard that increasing the scope of analysis through the molding of specialized functionalities for the detection of hallucinations like measurements for the amount of accurate information the logical consistency of the narrative and how far it strays from facts deepens the analysis the model could deliver. In these ways a comprehensive methodology allows a more complete assessment of the credibility of such content.

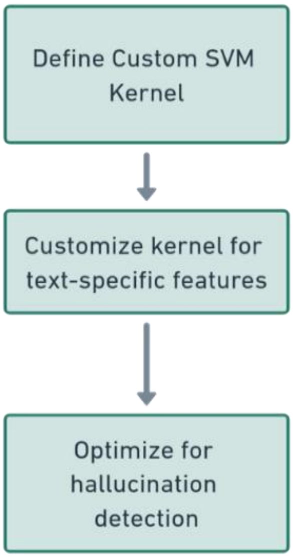
Define Custom SVM Kernel
Customize kernel for text-specific features
Optimize for hallucination detection

Figure 4. SVM classifier training

This can be done much better by tailoring the kernel in the SVM to the peculiar characteristics of textual data. In this way it can exploit a text-similarity metric or directly integrate constructed features into a kernel for increased discriminative power of an SVM between original and faked texts based on complicated patterns in the data. SVM classification requires training a model on a labeled as can be seen in Figure 4.

Dataset that differentiates between accurate and hallucinated texts. This involves the proper selection of kernel to be it linear RBF or custom optimization of extra hyperparameters like the regularization parameter and in the case of RBF the kernel coefficient. The experiments are conducted to obtain the best setting as the choice of kernel and parameters has a significant effect on performance. Evaluation of the model is necessary to understand its efficiency. Accuracy, precision, recall, and F1 score are some of the metrics which provide different perspectives towards performance including overall correctness ability to correctly classify hallucinations, and trade-off between precision and recall. K-fold cross validation strengthens it by testing the model on subsets of the data.

3.3 Model training and validation

Enhancing the model as seen in Figure 5 and Figure 6 is a repetitive procedure that entails modifying the feature set and SVM parameters according to evaluation feedback. Possible approaches to improve accuracy and generalizability to unfamiliar texts include optimising feature extraction processes adjusting the SVM kernel or augmenting the dataset.

Characterized by the incorporation and implementation of developed ideas this transition step from development into practical use is ensured by integrating a trained model into NLP pipelines or applications. This will guarantee consistent and very accurate processing of new texts to identify hallucinations. Through this means, users or systems can effectively tell between authentic and counterfeit content in real-life situations can be seen in Figure 7 and Figure 8.

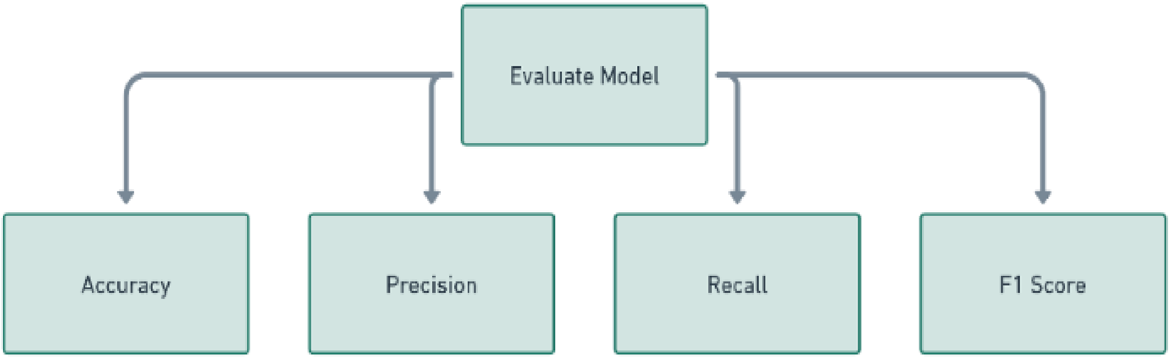

Figure 5. Model evaluation

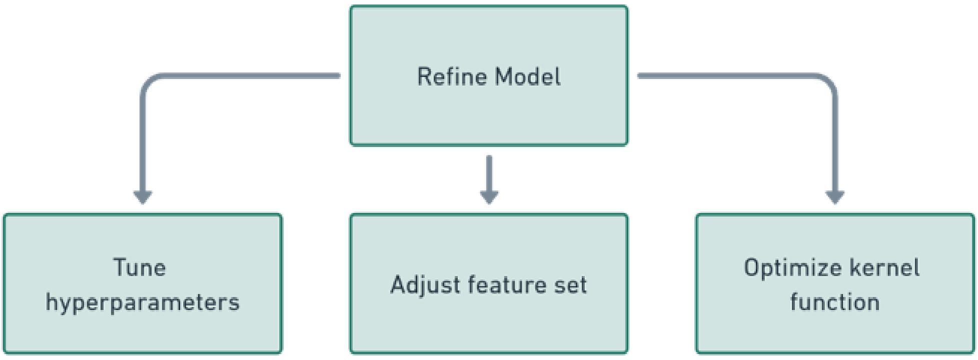

Figure 6. Model refinement

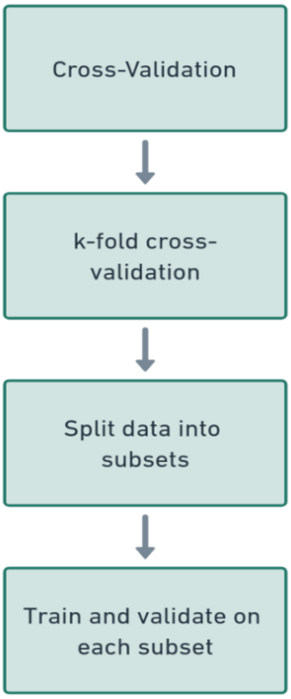

Figure 7. Cross-validation

The formal derivation that makes explicit how the hybrid kernel integrates lexical and deep-semantic similarity based from references [21-26]. Let $\phi[BERT]^{(\chi)} \in \mathbb{R}^{768}$ the sentence-level embedding of document χ obtained from a pretrained BERT encoder. $V_{TFIDF}(\chi)$ shows the sparse TF-IDF vector, and define the hybrid semantic kernel as the following:

$$K_{hybrid}(\chi, \chi') = \lambda_1 \cos\big(\phi BERT(\chi), \phi BERT(\chi')\big) + \lambda_2 exp[-\gamma\ ||\phi BERT(\chi), \phi BERT(\chi')||^2] + \lambda_3 (vTFIDF(\chi), vTFIDF(\chi'))$$

where, $\gamma > 0$ controls the bandwidth of the radial term.

- Term 1 — cosine kernel

cos() measures angular proximity between BERT embeddings and emphasising deep-semantic similarity.

- Term 2 — RBF kernel

exp $[-\gamma|| \ ||^2]$ that imposes a local smoothness prior in the same embedding space and mirroring the classical RBF kernel.

- Term 3 — linear kernel

$(vTFIDF(\chi), vTFIDF(\chi'))$ captures lexical overlap through the standard inner product on sparse TF-IDF features.

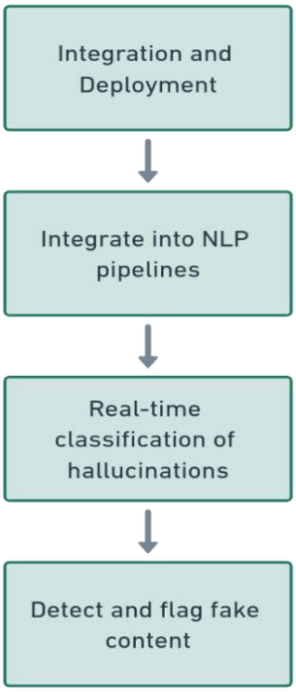
Integration and Deployment
Integrate into NLP pipelines
Real-time classification of hallucinations
Detect and flag fake content

Figure 8. Deployment

3.4 Detection of hallucinations

Labeling protocol is a passage contains "hallucinated" facts that cannot be verified or that contradict the prompt or context and it is considered hallucinated or if not, it is considered accurate. Two annotators independently labeled each instance and adjudication was used to settle disputes. Prediction workflow as previously mentioned features are extracted from a new text then run through the trained SVM and the class with the highest decision score is determined.

Table 1. Comparison of non-hallucinated vs hallucinated datasets

Aspect	Non-Hallucinated Dataset	Hallucinated Dataset
Language Style	Technical and precise, with occasional metaphors.	Whimsical and fantastical, full of metaphors and magical realism.
Content Focus	Accurate description of OSI model functions and network devices.	Surreal descriptions of network behavior with mythical elements.
Repetition	Moderate repetition of technical terms and functions.	Frequent repetition, especially of fantastical descriptions.
Purpose	Educational, focused on explaining network behavior.	Creative and abstract, likely for entertainment or illustrative purposes.
Accuracy	High accuracy, factual representation of networking concepts.	Low accuracy, imaginative rather than fact-based.

Table 2. Random samples from non-hallucinated and hallucinated datasets

Non-Hallucinated Samples	Hallucinated Samples
Routers pass data packets between networks based on Layer 3 addresses and make routing decisions.	The secret power of hubs lies in their ability to sing to the bytes.
Bridges create separate collision domains to increase network bandwidth and make forwarding decisions based on MAC addresses.	The Application layer is rumored to contain a portal to another dimension where lost data packets end up.

Network Interface Cards (NICs) connect end-user devices to a network and have a unique MAC address.	Hubs, the gossipers of the network, share everything they hear with everyone, without discretion.
Repeaters regenerate network signals to allow them to travel longer distances on the medium.	In the OSI model, the Data Link layer magically transforms data packets into dragons.
Transport Layer manages data segmentation from the sending host and reassembly at the receiving host.	Routers consult an ancient map, called the routing table, to decide the fate of each data packet.
The Physical Layer defines specifications for the physical link between end systems, including voltage levels and connectors.	In the OSI model, the Physical layer magically transforms data packets into dragons.
Session Layer establishes, manages, and terminates sessions between communicating hosts.	Mystical creatures known as 'network administrators' wield the power to configure switches with arcane commands.
Application Layer provides services directly to user's applications, not to any other OSI layer.	Switches have a hidden mode where they can teleport data packets instantly across the network.
Data Link Layer ensures reliable transit of data across a physical link, dealing with addressing and error notification.	The Data Link layer is rumored to contain a portal to another dimension where lost data packets end up.
The OSI Model consists of seven layers, each with specific functions for network communication.	In the OSI model, the Network layer magically transforms data packets into unicorns.
Routers consult an ancient map, called the routing table, to decide the fate of each data packet.	The secret power of switches lies in their ability to cast spells on data.
Bridges, unlike their mythical counterparts, do not require a troll's permission to pass data packets.	Legend has it that the Presentation layer of the OSI model was designed by a wizard from the ancient times.
Network Layer is responsible for connectivity and path selection between geographically separated networks.	Bridges, unlike their mythical counterparts, do not require a troll's permission to pass data packets.
Hubs, the gossipers of the network, share everything they hear with everyone, without discretion.	The secret power of routers lies in their ability to sing to the bytes.
The OSI Model consists of seven layers, each with specific functions for network communication.	Switches have a hidden mode where they can teleport data packets instantly across the network.

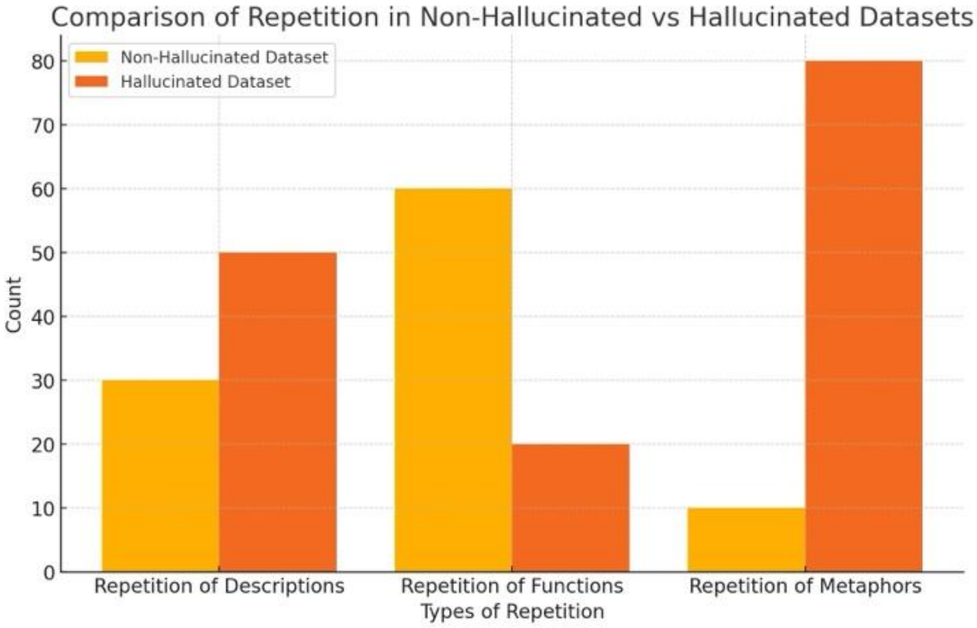

Figure 9. Summary of dataset analysis

The Appendix demonstrates that even accurate texts have a higher density of factual nouns and citation markers and hallucinated texts have higher frequencies of speculative verbs, metaphorical phrases and unverifiable named entities. The observed contrasts support the use of embedding-based features and validate the description of hallucination detection as a semantic classification task in Tables 1-2 and Figure 9.

4. RESULTS

This section presents and discusses the results of the implemented models, including Custom Kernel SVM, Logistic Regression, Random Forest and Naïve Bayes. The comparison of models is done using standard evaluation metrics, such as accuracy, precision, recall and F1-score.

The Custom Kernel SVM model performed exceptionally well in classification, and its accuracy rate was 98.57% as shown in Table 3 and Figure 10.

Table 3 shows how Custom Kernel SVM has an almost perfect score for classes with 0.99 F1-score for the larger class. The small reduction in recall for class 0 is suggestive of some misclassification but in general the model is robust. And the high values for accuracy and for metrics on performance depict SVM to effectively recognize basic data patterns.

The Logistic Regression model produced 93.81% accuracy that have achieved perfect recall score but with somewhat lower precision compared to Custom Kernel SVM. The logistic regression confusion matrix and performance are shown in Table 3 and Figure 11.

Table 3. The performance of the modes

Model	Accuracy	Macro-Precision	Macro-Recall	Macro-F1
Hybrid-Kernel SVM	98.6%	0.99	0.96	0.97
Random Forest	97.1%	0.78	0.86	0.82
Naïve Bayes	97.1%	0.78	0.86	0.82
Logistic Regression	93.8%	0.53	0.97	0.68

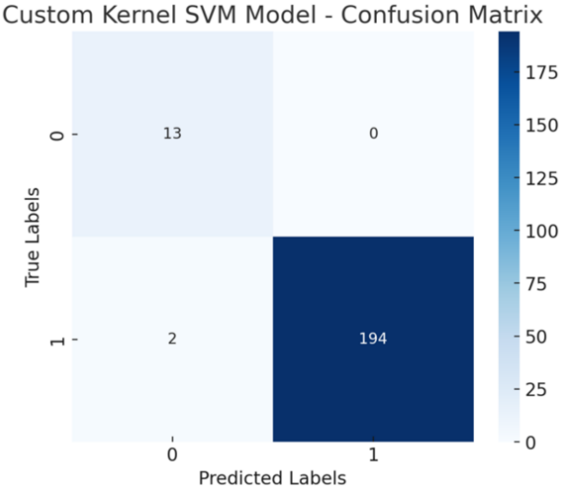

Figure 10. Confusion matrix of Custom Kernel SVM model

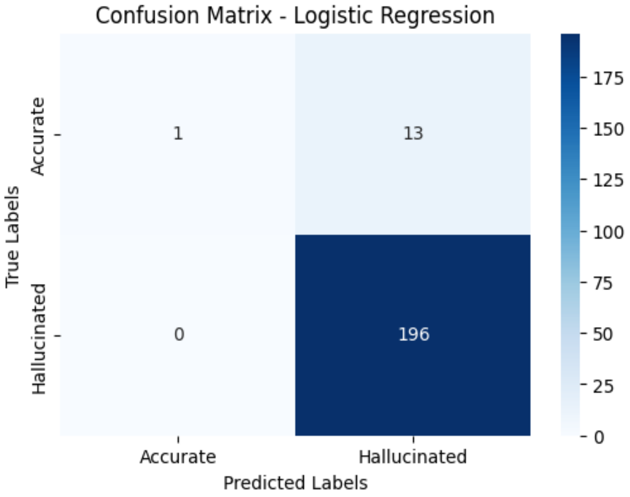

Figure 11. Confusion matrix of logistic regression

Although overall accuracy is good but deterioration in precision on the margin implies that Logistic Regression may have incorrectly labelled some events as false positives and thus have produced lower overall precision.

The Random Forest model did exceptionally well and having an accuracy rate of 97.14% and narrowly lagging behind Custom Kernel SVM as shown in Table 3 and Figure 12.

Random Forest model shows a good recall and accuracy that indicating good robustness. The capability of the system to specify fine-grained decision boundaries is an improvement but the minute loss in accuracy to Custom Kernel SVM identifies SVM to be better for the dataset.

While Naïve Bayes that implemented record a 97.14% accuracy that is comparable to Random Forest and Naïve Bayes under independent feature assumption may however produce overly general assumption on data distributions and hence compromise on precision and recall.

Table 1 and Figure 13 show the performance of Naïve Bayes.

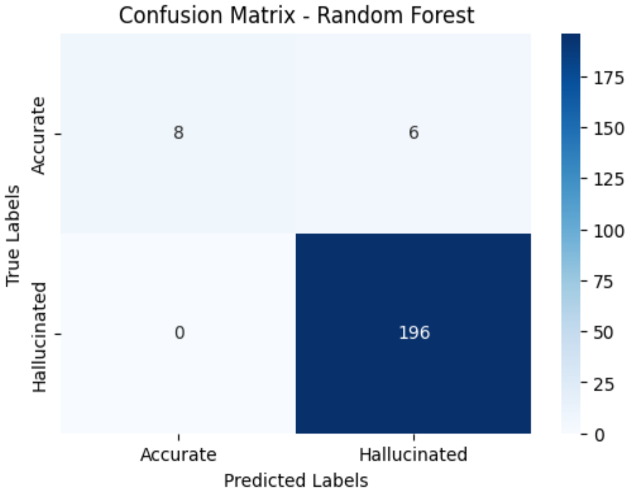

Figure 12. Confusion matrix of Random Forest

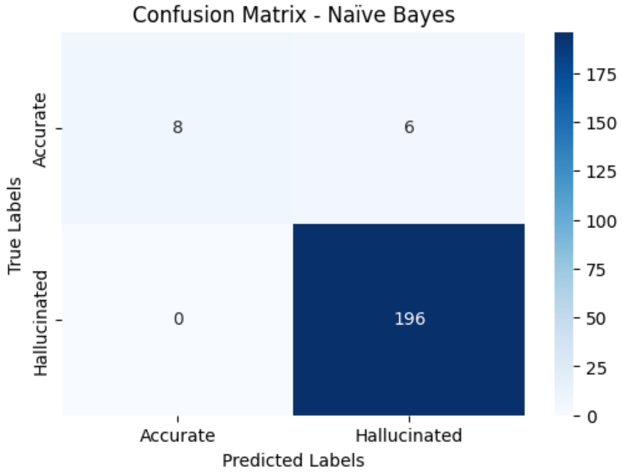

Figure 13. Confusion matrix of Naïve Bayes model

The Custom Kernel SVM performed better than other classifiers in regard to accuracy and general classification metrics. The Naïve Bayes and Random Forest models did adequately and their precision and accuracy, however, fell behind only by a small amount compared to proposed model. The Logistic Regression even it successful registered the worst result which indicating that linear decision boundaries may fall short in the capturing dataset's complexity.

The exceptional accuracy and precision of Custom Kernel SVM make it the perfect choice for such a classification. Its capability to discriminate effectively between classes, and most importantly in regions of high dimensions, makes it a top contender.

Random Forest and Naïve Bayes are good competitors, having better recall and reliability. Nonetheless, their precision is somewhat lower than Custom Kernel SVM. Logistic Regression is clear and to the point but its lesser precision means that it may not always be the perfect solution for multifaceted classification problems.

A false negative spreads false information to end users while a false positive incurs needless human review expenses in high-stakes applications. The hybrid-kernel SVM achieves the ideal trade-off completely removing false positives while keeping false negatives below 1%, as shown in Table 3 and Figures 10-13.

There is a notable difference between recall 0.97 and precision 0.53 in logistic regression that suggesting a high rate of false positives. While Random Forest and Naïve Bayes close this gap precision = 0.78 and still fall the SVM by 21 percentage points. Accuracy alone understates baseline performance and balanced metrics confirm the semantic kernel's superiority. For output pipelines removing false positives is essential because every flagged passage requires manual fact-checking. The two false negatives result from borderline cases that contain obscure but verifiable facts and this type of error is less expensive than allowing fabricated content to remain undetected.

5. CONCLUSIONS AND FUTURE WORK

The developed Custom Kernel SVM-based hallucination detection system has demonstrated a significant capacity for discerning accurate from hallucinated AI-generated text. In conclusion, Custom Kernel SVM is the optimal model for the classification and it is better than Logistic Regression, Random Forest and Naïve Bayes in most of the crucial metrics which those models have been implemented. Naïve Bayes and Random Forest do have good options available to them but they cannot match SVM's precision and overall accuracy. Logistic Regression though understandable could consider less suitable. A domain-specific dataset has the possibility of kernel weight overfitting and a limited review of real-time deployment latency are among the limitations. For a future work, several some points will be considered, such as firstly expand to multilingual and cross-domain corpora secondly exploring the learnable or adaptive kernels or combine kernel techniques with entropy- or tool-based detectors and lastly evaluate integration in retrieval-augmented and interactive AI systems.

REFERENCES

[1] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., Fung, P. (2022). Survey of hallucination in natural language generation. arXiv preprint arXiv:2202.03629. https://doi.org/10.48550/arxiv.2202.03629

[2] Safar, D., Safar, M., Jaafar, S., Al-Yachli, B.A., Shukri, A.K., Rasheed, M.H. (2025). Hallucinations in GPT-2 trained model. Ingénierie des Systèmes d'Information, 30(1): 31-41. https://doi.org/10.18280/isi.300104

[3] Farquhar, S., Kossen, J., Kuhn, L., Gal, Y. (2024). Detecting hallucinations in large language models using semantic entropy. Nature, 630(8017): 625-630. https://doi.org/10.1038/s41586-024-07421-0

[4] Su, W., Wang, C., Ai, Q., Hu, Y., Wu, Z., Zhou, Y., Liu, Y. (2024). Unsupervised real-time hallucination detection based on the internal states of large language models. In Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, pp. 14379-14391. https://doi.org/10.18653/v1/2024.findings-acl.854

[5] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. arXiv preprint arXiv:2005.11401. https://doi.org/10.48550/arxiv.2005.11401

[6] Nguyen, T.H.B., Tran, T.D.H. (2023). Exploring the efficacy of ChatGPT in language teaching. AsiaCALL Online Journal, 14(2): 156-167. https://doi.org/10.54855/acoj.2314210

[7] Athaluri, S.A., Manthena, S.V., Kesapragada, V.K.M., Yarlagadda, V., Dave, T., Duddumpudi, R.T.S. (2023). Exploring the boundaries of reality: Investigating the phenomenon of artificial intelligence hallucination in scientific writing through ChatGPT references. Cureus, 15(4): e37432. https://doi.org/10.7759/cureus.37432

[8] Deuze, M., Beckett, C. (2022). Imagination, algorithms and news: Developing AI literacy for journalism. Digital Journalism, 10(10): 1913-1918. https://doi.org/10.1080/21670811.2022.2119152

[9] Qiu, Y., Ziser, Y., Korhonen, A., Ponti, E.M., Cohen, S.B. (2023). Detecting and mitigating hallucinations in multilingual summarisation. arXiv preprint arXiv:2305.13632. https://doi.org/10.48550/arXiv.2305.13632

[10] Altınel, B., Ganiz, M.C., Diri, B. (2015). A corpus-based semantic kernel for text classification by using meaning values of terms. Engineering Applications of Artificial Intelligence, 43: 54-66. https://doi.org/10.1016/j.engappai.2015.03.015

[11] Xu, W., Agrawal, S., Briakou, E., Martindale, M.J., Carpuat, M. (2023). Understanding and detecting hallucinations in neural machine translation via model introspection. arXiv preprint arXiv:2301.07779. https://doi.org/10.48550/arXiv.2301.07779

[12] Liu, T., Zhang, Y., Brockett, C., Mao, Y., Sui, Z., Chen, W., Dolan, B. (2021). A token-level reference-free hallucination detection benchmark for free-form text generation. arXiv preprint arXiv:2104.08704. https://doi.org/10.48550/arXiv.2104.08704

[13] Sadat, M., Zhou, Z., Lange, L., Araki, J., Gundroo, A., Wang, B., Menon, R.R., Parvez, M.R., Feng, Z. (2023). DelucionQA: Detecting hallucinations in domain-specific question answering. In Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, pp. 822-835. https://doi.org/10.18653/v1/2023.findings-emnlp.59

[14] Badathala, N., Bhattacharyya, P. (2023). Detection of Rare Language Phenomena in NLP-Hallucination, Hyperbole, and Metaphor: A Survey. CFILT, IIT Bombay.

[15] Guerreiro, N.M., Alves, D.M., Waldendorf, J., Haddow, B., Birch, A., Colombo, P., Martins, A.F.T. (2023). Hallucinations in large multilingual translation models. Transactions of the Association for Computational Linguistics, 11: 1500-1517. https://doi.org/10.1162/tacl_a_00615

[16] Chen, X., Wang, C., Xue, Y., Zhang, N., Yang, X., Li, Q., Shen, Y., Gu, J., Chen, H. (2024). Unified hallucination detection for multimodal large language models. arXiv preprint arXiv:2402.03190. https://doi.org/10.48550/arxiv.2402.03190

[17] Salman, A.M., Abbas, H.H., Al Sayed, I.A.M.,

Abdulbaqi, A.S., Sekhar, R., Shah, P., Sandbhor, S. (2024). Enhanced support vector machine-based intelligent classification of trusted nodes in WBAN for resilient infrastructure. Mathematical Modelling of Engineering Problems, 11(8): 2267-2274. https://doi.org/10.18280/mmep.110829

[18] Jasim, N.A., Ibrahim, A.A., Hatem, W.A. (2023). Leveraging support vector machine for predictive analysis of earned value performance indicators in Iraq's oil projects. Mathematical Modelling of Engineering Problems, 10(6): 2003-2013. https://doi.org/10.18280/mmep.100610

[19] Tsaneva, D., Shao, J. (2018). Assisting investors with collective intelligence. In Proceedings of the First International Conference on Data Science, E-learning and Information Systems, New York, NY, United States, pp. 1-6. https://doi.org/10.1145/3279996.3280030

[20] Agha, R.A.A.R., Sefer, M.N., Fattah, P. (2018). A comprehensive study on sign languages recognition systems using (SVM, KNN, CNN and ANN). In Proceedings of the First International Conference on Data Science, E-learning and Information Systems, New York, NY, United States, pp. 1-6. https://doi.org/10.1145/3279996.3280024

[21] Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3): 273-297. https://doi.org/10.1007/bf00994018

[22] Devlin, J., Chang, M., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Human Language Technologies, Minneapolis, Minnesota, pp. 4171-4186. https://doi.org/10.18653/v1/n19-1423

[23] Reimers, N., Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084. https://doi.org/10.48550/arxiv.1908.10084

[24] Lodhi, H., Shawe-Taylor, J., Cristianini, N., Watkins, C.J.C.H. (2000). Text classification using string kernels. Journal of Machine Learning Research, 2: 419-444.

[25] Salton, G., Wong, A., Yang, C.S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11): 613-620. https://doi.org/10.1145/361219.361220

[26] Schölkopf, B., Smola, A., Müller, K. (1998). Nonlinear component analysis as a kernel Eigenvalue problem. Neural Computation, 10(5): 1299-1319. https://doi.org/10.1162/089976698300017467

APPENDIX

A. Hallucination Detection

It first loads the SVM model and a feature vectorizer from file paths provided to these variables. This is a crucial step as it guarantees the same parameters and feature space used while training the model in the classification process, impacting consistency and accuracy in predictions. A function to read the input CSV file. It can handle the fluctuation of real-world data especially in terms of encoding formats. This approach very effectively handles common obstacles in the handling of text data from different sources ensuring that it loads the data correctly into a panda’s frame for further manipulation. If the text data is loaded successfully then the script proceeds to iterate over each text entry. Before classification each text undergoes various preprocessing stages involving cleaning and then tokenization. This step in the text is for the sake of standardizing the format of the text and reducing unwanted interference that may impact classification. After the preprocessing, the text will be turned into a feature vector with the loaded vectorizer. This step is important in ensuring that the input structure to the model is as expected. The SVM to relatively accurately predict the classification of a text as either authentic or counterfeit. An appropriate predictor of class is created by applying learned decision boundaries for a SVM which have been defined using a training dataset containing labeled data during model training. Finally, it concatenates the original text data and their respective predictions into one data frame. After this data frame is written as a CSV file. This operation creates a record of all classifications for the texts. This resulting CSV file is one concrete output of running this script that can help in gaining insight into how well the model thinks it has scored on each of the texts.

It also makes manifest the flexibility of the system in handling textual data. It represents the reading of an input file in various encodings thus able to cope with real-world data presented in multiple formats robustly and efficiently. This brings forth an automated feature extraction and prediction by allowing the integration of specialized preprocessing and feature extraction with SVM prediction for text data classification dispensing manual feature engineering. The system possesses the capacity to effectively handle and forecast extensive datasets while also storing the outcomes for subsequent analysis or utilisation in applications that necessitate accurate textual content, which can be seen in Tables 1-2 and Figure 9.