Real-Time Explainable Spam Message Classification Using DistilBERT

Tanaya Chaudhari[1], Nachiket Shewale[2], Siddhesh Jadhav[3], Atharva Chavan[4] Prof. Vijaya Patil[5]
[1,2,3,4]*School of Computing, MIT Art, Design and Technology University, Pune, India*
[5]*Research Guide: Prof, 4School of Computing, MIT Art, Design and Technology University, Pune, India*

Abstract— **The rapid growth of digital communication technologies, including email services, mobile messaging applications, and social networking platforms, has significantly increased the volume of spam messages transmitted across the internet. Spam messages are unsolicited communications that often contain phishing links, fraudulent advertisements, deceptive promotional offers, or malicious content designed to exploit users. These messages not only create inconvenience but also pose serious cybersecurity threats such as identity theft, financial loss, and unauthorized access to sensitive information. Traditional spam detection techniques, including rule-based filtering and classical machine learning algorithms such as Naive Bayes and Support Vector Machine, have been widely used to address this issue. However, these approaches are limited in their ability to capture contextual relationships within text and often fail to detect sophisticated spam messages that use semantic manipulation and obfuscation techniques.**

To overcome these limitations, this research proposes a real-time explainable spam message classification system based on the DistilBERT transformer architecture. DistilBERT is a lightweight and computationally efficient version of BERT that maintains strong contextual language understanding while reducing model complexity and processing overhead. The model is fine-tuned using the PyTorch deep learning framework in combination with the Hugging Face Transformers library to classify incoming messages as spam or legitimate. Furthermore, to enhance transparency and interpretability, explainable artificial intelligence techniques such as SHAP and LIME are integrated into the system to identify and visualize the words that contribute most significantly to the model's predictions.

For practical implementation, the trained model is deployed using a FastAPI-based backend system and hosted on cloud platforms such as Amazon Web Services and Render, enabling scalable and real-time spam detection through API-based communication. Experimental evaluation conducted on the SMS Spam Collection Dataset demonstrates that the proposed system achieves high classification performance with an accuracy of 96

Index Terms— **DistilBERT, Explainable Artificial Intelligence (XAI), FastAPI, Natural Language Processing (NLP), Spam Detection, Text Classification, Transformer Models.**

I. INTRODUCTION

The rapid advancement of digital communication technologies has significantly transformed the way individuals and organizations exchange information. Platforms such as email services, mobile messaging applications, and social networking systems have become essential tools for communication in both personal and professional environments. Every day, billions of messages are transmitted across these platforms, enabling fast and convenient interaction across geographical boundaries. However, this widespread adoption has also led to a substantial increase in unsolicited and malicious messages, commonly referred to as spam. Spam messages typically include phishing links, fraudulent advertisements, deceptive promotional offers, and misleading financial schemes designed to manipulate users and exploit sensitive information. As a result, spam has evolved from a minor inconvenience into a serious cybersecurity threat, leading to issues such as identity theft, financial loss, and unauthorized access to confidential data.

Over the years, various techniques have been developed to detect and filter spam messages. Traditional approaches primarily rely on rule-based filtering mechanisms and classical machine learning algorithms such as Naive Bayes and Support Vector Machine. These methods analyze predefined keywords or statistical patterns within messages to

classify them as spam or legitimate. Although these techniques are computationally efficient and relatively easy to implement, they suffer from significant limitations. In particular, they are unable to effectively capture contextual relationships within text and often fail to detect sophisticated spam messages that employ semantic manipulation, word obfuscation, or disguised language patterns to bypass detection systems.

Recent advancements in Natural Language Processing (NLP) have introduced transformer-based models that significantly improve text classification performance by capturing contextual meaning within language. Among these models, BERT (Bidirectional Encoder Representations from Transformers) has demonstrated state-of-the-art results in various NLP tasks. However, its large size and high computational requirements make it less suitable for real-time applications. To address this limitation, DistilBERT, a lightweight and efficient variant of BERT, has been developed using knowledge distillation techniques. DistilBERT retains most of the contextual understanding capabilities of BERT while reducing computational complexity, making it more suitable for real-time spam detection systems.

Despite improvements in classification accuracy, another major challenge in modern spam detection systems is the lack of interpretability. Many deep learning models operate as black-box systems, providing predictions without clear explanations, which reduces user trust and system transparency. To overcome this issue, the field of Explainable Artificial Intelligence (XAI) has introduced techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) that help interpret model predictions by identifying important features influencing the output.

In this research, a real-time explainable spam message classification system is proposed using the DistilBERT transformer architecture. The model is finetuned using the PyTorch deep learning framework and the HuggingFace Transformers library to accurately classify messages as spam or legitimate. To enhance transparency, SHAP and LIME techniques are integrated to provide interpretable insights into the model's decisions. Furthermore, the system is deployed using a FastAPI-based backend and cloud platforms such as Amazon Web Services and Render, enabling scalable and real-time spam detection. The proposed approach aims to achieve high classification accuracy while ensuring efficiency, interpretability, and practical applicability in modern digital communication environments.

II. PROBLEM DEFINITION

The rapid expansion of digital communication technologies has significantly increased the volume of messages exchanged through platforms such as email services, mobile messaging applications, and social networking systems. While these platforms enable efficient and instant communication, they have also become a major channel for the distribution of spam messages. Spam refers to unsolicited messages sent in bulk without the consent of the recipient, often containing phishing links, fraudulent advertisements, misleading promotional offers, or malicious attachments.

These messages are designed to deceive users into revealing sensitive information such as login credentials, banking details, or personal data. Consequently, spam has emerged as a critical cybersecurity concern, affecting both individuals and organizations by causing financial loss, privacy breaches, and unauthorized access to confidential information.

One of the primary challenges in spam detection is the constantly evolving nature of spam techniques. Cybercriminals continuously develop new strategies to evade existing filtering systems. These techniques include word obfuscation, deliberate misspellings, use of special characters, and contextual manipulation of language. For instance, common spam keywords such as "free" or "win" may be modified into "fr33" or "w!n" to bypass keyword-based filters. Additionally, spam messages are often designed to closely resemble legitimate communications from trusted sources such as banks, e-commerce platforms, or government organizations. This makes it increasingly difficult for automated systems to accurately distinguish between genuine and malicious messages.

Traditional spam detection approaches, including rule-based filtering and classical machine learning algorithms such as Naive Bayes and Support Vector Machine, rely heavily on predefined rules or frequency-based features. Although these methods are computationally efficient, they fail to capture deeper contextual relationships within text. As a result, they are unable to effectively detect complex spam

messages that depend on semantic variation or contextual manipulation. Furthermore, these approaches often require extensive feature engineering and manual preprocessing, which limits their adaptability to evolving spam patterns.

Recent deep learning–based approaches have improved classification accuracy by learning complex patterns directly from data. However, many of these models operate as black-box systems, providing predictions without clear explanations. This lack of interpretability reduces user trust and makes it difficult to validate the model's decisions, especially in securitycritical applications.

Therefore, there is a need for an advanced spam detection system that can accurately identify sophisticated spam messages by understanding contextual language patterns, provide interpretable predictions to enhance transparency, and support real-time deployment for practical use in modern communication platforms.

III. OBJECTIVES

The primary objective of this research is to design and develop an intelligent, efficient, and interpretable spam message classification system capable of accurately identifying spam messages in modern digital communication environments. With the rapid growth of messaging platforms and increasing cybersecurity threats, it is essential to create a system that not only improves detection accuracy but also understands the contextual meaning of textual data and provides transparent decision-making. The proposed framework focuses on combining advanced natural language processing techniques with explainable artificial intelligence to achieve reliable and real-time spam detection.

The specific objectives of this research are as follows:

1. To design and develop an automated spam message classification system that can effectively distinguish between spam and legitimate (ham) messages across various communication platforms such as SMS, email, and social media.
2. To utilize the DistilBERT transformer-based model to capture contextual relationships between words in a message, enabling the system to detect complex and sophisticated spam patterns that traditional methods fail to identify.
3. To integrate explainable artificial intelligence techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) in order to interpret the model's predictions and identify the most influential words contributing to classification results.
4. To enhance transparency and user trust by providing clear and understandable explanations for each classification decision generated by the system.
5. To develop a scalable backend architecture using FastAPI that supports real-time spam detection through API-based communication with external applications.
6. To deploy the proposed system on cloud platforms such as Amazon Web Services or Render, ensuring scalability, accessibility, and efficient processing of large volumes of messages.
7. To evaluate the performance of the proposedmodel using standard classification metrics including accuracy, precision, recall, and F1-score in order to ensure reliability and effectiveness.

These objectives collectively aim to create a robust, accurate, interpretable, and scalable spam detection system suitable for real-world applications.

IV. LITERATURE REVIEW

Spam detection has been an important research area due to the rapid growth of digital communication platforms such as email systems, mobile messaging applications, and social media networks. With the increasing volume of online communication, the number of spam messages has also increased significantly. These messages often contain phishing links, misleading advertisements, or malicious content that may cause financial loss or data breaches. To address this issue, researchers have proposed various techniques using machine learning, deep learning, and natural language processing methods.

4.1. Naive Bayes-Based Spam Detection

One of the earliest approaches for spam detection is the Naive Bayes classifier. This probabilistic model classifies messages based on the likelihood of words appearing in spam or legitimate messages. It assumes independence between words and calculates conditional probabilities to determine classification. Due to its simplicity, low computational cost, and fast

execution, Naive Bayes became widely used in early spam filtering systems. However, the assumption of independence between words is unrealistic in natural language, as words often depend on each other contextually. As a result, Naive Bayes struggles to capture complex linguistic patterns and fails to detect advanced spam messages that rely on semantic manipulation.

4.2. Support Vector Machine (SVM) Approaches

Support Vector Machine (SVM) is another widely used supervised learning algorithm for spam classification. SVM works by finding an optimal hyperplane that separates spam and non-spam messages in a high dimensional feature space. Text data is usually transformed into numerical vectors using techniques such as bag-of-words or TF-IDF. SVM models have shown better accuracy than simple probabilistic models due to their ability to handle high-dimensional data effectively. However, SVM requires extensive feature engineering and preprocessing, which increases system complexity. Additionally, SVM does not effectively capture sequential or contextual relationships between words in longer text messages.

4.3. Deep Learning Approaches Using RNN and LSTM

With advancements in artificial intelligence, deep learning techniques such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks have been applied to spam detection. These models are designed to process sequential data and capture dependencies between words in a sentence. LSTM networks, in particular, can retain long-term dependencies and learn contextual relationships within text. This allows them to achieve higher classification accuracy compared to traditional machine learning models. However, training RNN and LSTM models requires significant computational resources and time. Additionally, these models process text sequentially, which limits their efficiency when dealing with large datasets or real-time applications.

4.4. Transformer-Based Models and BERT

Recent developments in Natural Language Processing introduced transformer-based architectures that significantly improved language understanding capabilities. Unlike sequential models, transformers use attention mechanisms to analyze relationships between all words in a sentence simultaneously. One of the most influential transformer models is BERT (Bidirectional Encoder Representations from Transformers). BERT processes text bidirectionally, meaning it considers both previous and next words to understand context. This enables the model to capture complex semantic relationships within language, resulting in state-ofthe-art performance in text classification tasks. However, BERT has a large number of parameters and requires high computational resources, making it less suitable for real-time applications.

4.5. DistilBERT for Efficient Text Classification

To overcome the limitations of large transformer models, DistilBERT was introduced as a lightweight and efficient version of BERT. It is developed using knowledge distillation, where a smaller model learns from a larger pretrained model. DistilBERT retains most of the contextual understanding capability of BERT while significantly reducing model size and computation time. This makes it suitable for real-time applications such as spam detection systems. Studies have shown that DistilBERT achieves comparable performance to BERT with improved efficiency, making it a strong candidate for deployment in practical systems.

4.6. Explainable Artificial Intelligence (XAI)

Although deep learning models provide high accuracy, they are often criticized for operating as blackbox systems. This lack of transparency makes it difficult to understand how predictions are made, reducing user trust. To address this issue, Explainable Artificial Intelligence (XAI) techniques have been introduced. XAI aims to interpret model predictions by identifying the most important features influencing the output. Two widely used XAI techniques are LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations). LIME explains individual predictions by approximating the model locally, while SHAP uses game theory concepts to assign importance values to features. These techniques improve transparency and make AI systems more trustworthy.

4.7. Summary of Existing Research

Existing research indicates that traditional machine learning methods such as Naive Bayes and SVM are

effective for detecting simple spam patterns but fail to capture contextual relationships in complex messages. Deep learning models such as RNN and LSTM improve performance but require high computational resources. Transformer-based models like BERT provide excellent contextual understanding but are computationally expensive. DistilBERT offers a balance between accuracy and efficiency, making it suitable for real-time applications. However, interpretability remains a challenge in deep learning models, which can be addressed using XAI techniques such as SHAP and LIME.

The proposed research builds upon these advancements by combining the efficiency of DistilBERT with the transparency of explainable AI techniques to develop a real-time, accurate, and interpretable spam detection system.

V. PROPOSED SYSTEM

The proposed system introduces a real-time explainable spam message classification framework designed to accurately identify spam messages using advanced techniques from Natural Language Processing (NLP). The system leverages the DistilBERT transformer based model for classification and integrates explainable artificial intelligence (XAI) techniques to enhance transparency. In addition to achieving high classification accuracy, the system is designed for real-time deployment using a scalable backend architecture.

The primary objective of the proposed system is to classify incoming text messages into two categories: spam and legitimate (ham). Unlike traditional approaches that rely on keyword matching or statistical features, the proposed method utilizes deep learning to analyze contextual relationships between words. This enables the system to detect sophisticated spam messages that may use semantic manipulation or disguised language patterns.

5.1. System Overview

The overall workflow of the proposed system begins with receiving a message from a user or an external application. The message undergoes preprocessing to remove noise and improve data quality. The cleaned text is then converted into tokenized inputs using the DistilBERT tokenizer. These tokens are passed to the fine-tuned DistilBERT model, which analyzes contextual relationships and generates a classification output. To improve interpretability, explainability techniques such as SHAP and LIME are applied to highlight the most influential words contributing to the prediction. Finally, the results are delivered through a FastAPIbased backend system.

5.2. System Modules

The proposed system is composed of several functional modules that work together to perform classification and explanation generation:

5.2.1. Data Input Module

This module is responsible for receiving input messages from users or external platforms such as SMS services, email systems, or social media applications. The input message is forwarded to the preprocessing stage.

5.2.2. Text Preprocessing Module

In this stage, the raw text is cleaned to remove unwanted elements such as punctuation, special characters, and extra spaces. The text is also normalized by converting it into lowercase format. This improves the quality of input data and enhances model performance.

5.2.3. Tokenization Module

The cleaned text is converted into numerical representations using the DistilBERT tokenizer. Tokenization splits the text into smaller units (tokens) and generates token IDs and attention masks required by the transformer model.

5.2.4. DistilBERT Classification Module

The tokenized input is passed to the fine-tuned DistilBERT model. The model uses self-attention mechanisms to analyze contextual relationships between words in the message. Based on this analysis, the model predicts whether the message is spam or legitimate.

5.2.5. Explainability Module

To improve transparency, the system integrates XAI techniques such as SHAP and LIME. These methods analyze the model's prediction and identify the words that contributed most to the classification result. This helps users understand why a message was classified as spam.

5.2.6. API and Backend Module

The classification results are processed through a FastAPI-based backend. This module enables

communication between the model and external applications. Users can send messages through API requests and receive predictions in real time.

5.2.7. Cloud Deployment Module

The system is deployed on cloud platforms such as Amazon Web Services or Render. Cloud deployment ensures scalability, reliability, and the ability to handle multiple requests efficiently.

5.3. System Working Flow

The operational workflow of the system can be summarized as follows:

1. A user or application sends a message to the system through an API request.
2. The message is processed by the preprocessingmodule.
3. The cleaned text is tokenized using the DistilBERT tokenizer.
4. The tokenized input is passed to the DistilBERTclassification model.
5. The model predicts whether the message isspam or legitimate.
6. Explainability techniques (SHAP/LIME) analyze the prediction.
7. The final result, along with explanation details,is returned through the API.

5.4. Advantages of the Proposed System

The proposed system offers several advantages over traditional spam detection approaches:

- High Accuracy: Uses transformer-based model for better contextual understanding.
- Real-Time Processing: FastAPI enables quick response for real-time applications.
- Interpretability: SHAP and LIME provide clear explanations for predictions.
- Scalability: Cloud deployment allows handling large volumes of data.
- Efficiency: DistilBERT reduces computational cost compared to BERT.

Overall, the proposed system combines advanced NLP techniques with explainable AI and scalable deployment to provide an efficient, accurate, and transparent solution for spam detection in modern communication systems.

VI. METHODOLOGY

The methodology of the proposed system describes the complete pipeline followed to design, develop, train, evaluate, and deploy the real-time explainable spam message classification framework. The primary goal of this methodology is to develop a system that not only achieves high classification accuracy but also ensures interpretability and real-time performance. The implementation is divided into multiple stages, including data collection, preprocessing, tokenization, model training, explainability integration, evaluation, and deployment. Each stage plays a critical role in improving the overall performance and reliability of the system.

6.1. System Workflow Overview

The overall workflow of the proposed system is illustrated in Fig. 1. The system begins by receiving input text messages, which are then preprocessed to remove noise and inconsistencies. The cleaned text is converted into tokenized representations and passed to the DistilBERT model for classification. The model predicts whether the message is spam or legitimate. Explainable AI techniques are then applied to interpret the prediction, and the final output is returned to the user through an API interface.

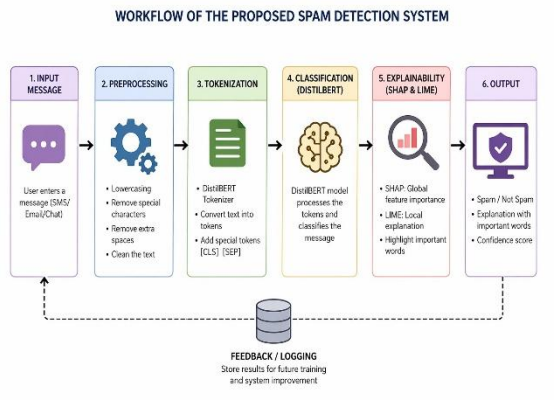

Fig. 1. Workflow of the Proposed Real-Time Explainable Spam Detection System

6.2. Data Collection

The first step in the methodology involves collecting a suitable dataset for training and evaluating the spam classification model. In this research, the SMS Spam Collection Dataset is utilized, which is widely recognized as a benchmark dataset in spam detection

research. This dataset consists of 5574 SMS messages collected from various real-world sources, including online forums, public datasets, and messaging platforms.

Each message in the dataset is labeled as either "spam" or "ham" (legitimate). Spam messages typically contain promotional offers, fraudulent schemes, or phishing attempts, whereas ham messages represent normal user communication. The dataset provides a balanced representation of both classes, enabling effective model training.

Before training, the dataset is divided into training and testing subsets. The training set is used to finetune the DistilBERT model, while the testing set is used to evaluate the model's performance on unseen data. This separation ensures that the model generalizes well and avoids overfitting.

6.3. Data Preprocessing

Text data collected from real-world sources often contains noise, including punctuation marks, special characters, inconsistent capitalization, and formatting irregularities. These issues can negatively impact model performance if not properly addressed.

To improve data quality, several preprocessing steps are applied:

- Lowercase Conversion: All text is converted into lowercase to ensure uniformity and reduce vocabulary size.
- Removal of Special Characters: Unnecessary symbols, punctuation, and non-text elements are removed.
- Whitespace Normalization: Extra spaces and formatting inconsistencies are eliminated.
- Text Cleaning: Irrelevant tokens and noise are removed to improve semantic clarity.

These preprocessing steps ensure that the input data is clean, consistent, and suitable for transformerbased models.

6.4. Tokenization

After preprocessing, the cleaned text must be converted into numerical form so that it can be processed by the neural network. This process is known as tokenization and is performed using the DistilBERT tokenizer.

Tokenization divides the input text into smaller units called tokens, which may represent words or subwords. Each token is mapped to a unique numerical identifier known as a token ID. In addition to token IDs, the tokenizer generates attention masks that indicate which tokens should be attended to by the model.

For example, a sentence such as "You have won a free prize" is broken into tokens that capture both individual words and subword structures. Padding is also applied to ensure uniform input length across all samples.

Tokenization plays a critical role in enabling the model to understand textual data in numerical form.

6.5. Model Training

The central component of the system is the DistilBERT model, which is fine-tuned for spam classification. DistilBERT is a transformer-based model that uses self-attention mechanisms to capture contextual relationships between words in a sentence.

The training process involves multiple steps:

1. The tokenized input data is fed into the DistilBERT model.
2. The model processes the input using attention mechanisms to analyze relationships between words.
3. A classification layer predicts whether the message is spam or ham.
4. The predicted output is compared with the actual label using a loss function.
5. Model parameters are updated using backpropagation.
6. The AdamW optimizer is used to improve training stability.
7. The process is repeated for multiple epochs untiloptimal performance is achieved.

Training is performed in a GPU-enabled environment to reduce computation time and improve efficiency.

6.6. Explainability Integration

Although deep learning models provide high accuracy, they often lack interpretability. To address this issue, the proposed system integrates Explainable Artificial Intelligence (XAI) techniques. Two techniques are used:

- LIME: Generates local explanations by approximating the model with a simpler interpretable model.

- SHAP: Uses Shapley values from game theory to measure the contribution of each feature to the prediction.

These methods identify important words in the message that influence the classification decision. For example, words like “free”, “offer”, and “win” may have a strong influence in classifying a message as spam.

Explainability improves transparency and helps users trust the system.

6.7. Model Evaluation

After training, the model is evaluated using standard classification metrics:

- Accuracy: Measures overall correctness of predictions.
- Precision: Measures correctness of spam predictions.
- Recall: Measures ability to detect all spam messages.
- F1 Score: Provides a balance between precision and recall.

These metrics are calculated using the confusion matrix, which summarizes prediction outcomes.

6.8. API Deployment

To enable real-time functionality, the trained model is integrated into a FastAPI-based backend system. FastAPI provides a high-performance interface for handling API requests.

The deployment process includes:

1. Receiving input message through API
2. Preprocessing and tokenization
3. Passing input to model
4. Generating prediction
5. Returning result to user

This enables real-time spam detection in practical applications.

6.9. Cloud Deployment

The final stage involves deploying the system on cloud platforms such as Amazon Web Services (AWS) or Render. Cloud deployment ensures scalability, reliability, and accessibility.

The system can handle multiple user requests simultaneously and provide quick responses. This makes it suitable for integration into real-world communication platforms.

6.10. Overall Methodology Flow

The complete methodology can be summarized as follows:

1. Data collection
2. Data preprocessing
3. Tokenization
4. Model training
5. Explainability integration
6. Model evaluation
7. API deployment
8. Cloud deployment

This structured approach ensures that the system is accurate, efficient, scalable, and interpretable.

VII. RESULTS

The experimental results evaluate the performance of the proposed spam detection system based on the DistilBERT model. The model was trained and tested using the SMS Spam Collection Dataset. The primary objective of the evaluation is to measure how accurately the system can classify messages as spam or legitimate (ham).

The dataset was divided into training and testing sets. The training data was used to fine-tune the model, while the testing data was used to evaluate its performance on unseen messages. The evaluation was performed using standard classification metrics such as accuracy, precision, recall, and F1-score.

7.1. Performance Metrics

The performance of the model is summarized in Table

Metric	Value
Accuracy	96%
Precision	95%
Recall	94%
F1 Score	94.5%

Table 1: Performance Metrics of Proposed Model

The results indicate that the proposed DistilBERTbased model achieves high classification accuracy. The precision value shows that most predicted spam messages are correct, while recall indicates that the model successfully detects most of the actual spam messages. The F1 score provides a balance between precision and recall.

7.2. Confusion Matrix

The confusion matrix provides a detailed view of the classification results.

	Predicted Spam	Predicted Ham
Actual Spam	705	42
Actual Ham	60	4767

Table 2: Confusion Matrix From the confusion matrix, it can be observed that

- 705 spam messages were correctly classified (True Positives)
- 4767 legitimate messages were correctly classi-fied (True Negatives)
- 42 spam messages were misclassified (FalseNegatives)
- 60 legitimate messages were misclassified (False Positives)

These results show that the model performs effectively with minimal classification errors.

7.3. Performance Graph

Fig. 2. Performance Metrics Representation

Accuracy: 96%
Precision: 95%
Recall: 94%
F1 Score: 94.5%

Fig. 2. Performance Metrics Representation

The graphical representation shows that all performance metrics have high values, indicating the reliability and effectiveness of the proposed model.

7.4. Discussion of Results

The experimental results demonstrate that the proposed DistilBERT-based spam detection system achieves strong performance across all evaluation metrics. The transformer-based architecture enables the model to understand contextual relationships between words, which improves its ability to detect complex spam messages.

Compared to traditional machine learning models, the proposed approach provides better accuracy and robustness. Additionally, the integration of explainable AI techniques enhances transparency by providing insights into the model's predictions.

Overall, the results confirm that the system is accurate, efficient, and suitable for real-time deployment in modern communication platforms.

VIII. CONCLUSION

The proposed research presents a real-time explainable spam message classification system based on the DistilBERT transformer architecture to address the growing challenges of spam in modern digital communication platforms. With the increasing use of email services, mobile messaging applications, and social media, the volume and complexity of spam messages have significantly increased, posing serious cybersecurity risks such as phishing attacks, financial fraud, and data breaches. Traditional spam detection methods, which rely on rule-based filtering and classical machine learning algorithms, often fail to capture contextual relationships within text and are ineffective against sophisticated spam techniques.

To overcome these limitations, the proposed system utilizes DistilBERT, a lightweight and efficient transformer-based model capable of understanding contextual language patterns. The model is finetuned using the PyTorch framework and the Hugging Face Transformers library to accurately classify messages as spam or legitimate. In addition to achieving high classification performance, the system integrates explainable artificial intelligence techniques such as SHAP and LIME to provide transparent and interpretable predictions. This enhances user trust and improves the usability of the system in real-world applications.

The experimental evaluation conducted using the SMS Spam Collection Dataset demonstrates that the proposed system achieves strong performance, with an accuracy of 96

Overall, the proposed system offers an efficient, accurate, and interpretable solution for spam detection, making it suitable for deployment in modern communication environments and contributing to improved cybersecurity and safer digital interactions.

REFERENCES

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019.

[2] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT: A distilled version of BERT: Smaller, faster, cheaper and lighter," in *Proc. NeurIPS Workshop*, 2019.

[3] M. Androutsopoulos, J. Koutsias, K. Chandrinos, and C. Spyropoulos, "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering," in *Proc. SIGIR*, 2000.

[4] H. Drucker, D. Wu, and V. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, 1999.

[5] T. Mikolov, M. Karafiat, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network-based language model," in *Proc. Interspeech*, 2010.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] M. Zhang and Y. LeCun, "Text understanding from scratch," in *Proc. NeurIPS*, 2015.

[8] M. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. ACM SIGKDD*, 2016.

[9] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, 2017.

[10] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," in *Proc. EMNLP*, 2020.

[11] Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[12] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[13] PyTorch Development Team, "PyTorch documentation," [Online]. Available: https://pytorch.org/docs

[14] Hugging Face, "Transformers library documentation," [Online]. Available: https://huggingface.co/docs/transformers

[15] Amazon Web Services, "AWS machine learning services documentation," [Online]. Available: https://aws.amazon.com/documentation

[16] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA, USA: O'Reilly Media, 2009.

[17] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999.

[18] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD*, 2016.

[19] F. Chollet, *Deep Learning with Python*. Shelter Island, NY, USA: Manning Publications, 2018.

[20] FastAPI, "FastAPI framework for building APIs," [Online]. Available: https://fastapi.tiangolo.com