Muthuraj A

se report_removed.pdf

Optimizing CVE Severity Prediction: A Hybrid TF-IDF and DistilBERT Approach with Attention-Based Deep Learning

CSE

Amrita Vishwa Vidyapeetham

Document Details

Submission ID

trn:oid:::1:3204692233

Submission Date

Apr 4, 2025, 1:16 PM GMT+5:30

Download Date

Apr 4, 2025, 1:19 PM GMT+5:30

File Name

se_report_removed.pdf

File Size

560.8 KB

9 Pages

3,002 Words

17,777 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

Optimizing CVE Severity Prediction: A Hybrid TF-IDF and DistilBERT Approach with Attention-Based Deep Learning

Abstract

Accurate prioritization of software vulnerabilities is the foundation of cybersecurity, and that is facilitated by correct forecasting of Common Vulnerabilities and Exposures (CVE) severity. This work introduces a state-of-the-art approach based on the integration of Term Frequency-Inverse Document Frequency (TF-IDF) and DistilBERT embeddings, which is further processed using an attention mechanism-based deep learning model for CVE severity score prediction. With a purified dataset from the National Vulnerability Database (NVD), our method addresses class imbalance via class weighting, yielding test accuracy at 93.52%, macro F1-score at 0.7926, and weighted F1-score at 0.9314. As opposed to the baseline work by Manjunatha et al. (2024), which used GPT-2 with accuracy at 84.2% and F1-score at 0.82. Its better performance can be attributed to its effective feature extraction and attention process for improving prediction consistency across severity classes. These findings demonstrate the capability of hybrid NLP methods in vulnerability assessment automation as a strong tool for risk management by cybersecurity experts.

Keywords: CVE Severity Prediction, TF-IDF, DistilBERT, Attention Mechanism, Deep Learning, Cybersecurity, NVD, Vulnerability Assessment, Class Imbalance, NLP.

INTRODUCTION:

The development of connected digital technologies has greatly expanded the reach of cybersecurity, most importantly to prevent software weaknesses which can cause data breaches, economic impacts, and reputational damage. The Common Vulnerabilities and Exposures (CVE) framework and Common Vulnerability Scoring System (CVSS) has devised a single, unified system in terms of reporting and scoring software and hardware system vulnerabilities. The CVSS allocates a score ranging from 0.0 to 10.0 and classifies the vulnerabilities as a function of severity levels—LOW (0.1–3.9), MEDIUM (4.0–6.9), HIGH (7.0–8.9), and CRITICAL (9.0–10.0)—depending on the exploitability and impact. The National Vulnerability Database (NVD), where these CVEs are hosted, however, enumerated more than 123,675 vulnerabilities by 2023, an exponential growth that makes manual severity assessment a growing impracticability. This massive volume, plus the sophistication of vulnerability descriptions, defines the paramount need for computer support tools with the ability to precisely and effectively predict severity in order to enact timely mitigation measure.

Conventional methods of evaluating severity in traditional history tend to be reliant on human examination by security professionals, not merely time-consuming but also subject to inconsistency by virtue of subjective judgment. Additionally, the written format of CVE descriptions—usually unstructured and heavy with technical terminology—is a major impediment for automated solutions. These descriptions hold significant information regarding the nature of the vulnerability, components impacted, and possible impact, but only meaningful patterns can be extracted using advanced natural language processing (NLP) techniques. Furthermore, the distribution of severity levels in the NVD dataset is strongly skewed towards an abundance of MEDIUM and HIGH severity vulnerabilities with no LOW and CRITICAL ones. This imbalance distorts model projections in such a fashion that it is not easy to clearly ascertain the most critical vulnerabilities that need to be addressed as a question of urgency.

To solve these problems, this project introduces a novel CVE severity prediction approach by incorporating Term Frequency-Inverse Document Frequency (TF-IDF) and DistilBERT embeddings into an attention-based deep learning model. TF-IDF extracts statistical text patterns with significant words highlighted for their importance, and DistilBERT, being a 66 million parameter distilled BERT model, offers contextual understanding with its pre-trained language expertise. Our attention component in our model also improves performance by concentrating on the most appropriate features in CVE descriptions to enhance prediction accuracy in every severity class. Our approach processes a cleaned NVD dataset, addressing class imbalance through class weighting for balanced learning. The model produced by the resulting model has a test accuracy of 93.52%, macro F1-score of 0.7926, and weighted F1-score of 0.9314, indicating the effectiveness in automating severity prediction.

The overall goal of this research is to create an automated and scalable solution towards CVE severity prediction, thus enabling better vulnerability management in cybersecurity. Our method, through automation, seeks to minimize the workload on security teams so that they can effectively allocate resources and prevent risks beforehand. The following sections of this report outline our approach, assess the performance of our model, and provide implications for future automated cybersecurity processes.

2. Related Work

Severity prediction of CVE has been a prime research area in the field of cybersecurity, and the methods have progressed significantly to accommodate the need of automating vulnerability scanning. Below is how the methods have transformed from the original conventional methods to the current NLP-based methods and presents the motivation for hybrid models.

2.1 Traditional Approaches

Early CVE severity prediction efforts relied mostly on manual analysis and rule-based approach. Security professionals would balance the characteristics of the vulnerabilities, such as exploitability and impact, based on pre-determined CVSS metrics such as Attack Vector, Attack Complexity, and Privileges Required. The metrics were used to calculate a base score, which itself was converted to the magnitude of severity. Some of the work applied statis36+tical methods, like linear regression, to estimate single CVSS values from pre-formatted data fields. Decision trees and fuzzy logic systems were some other conventional techniques, employing hand-tuned rules to transform vulnerability attributes to severity scores. While these methods established a formal paradigm, they were plagued by the disadvantage that they used manual feature engineering and could not cope with unstructured text-based data in CVE descriptions, resulting in scalability problems as the number of vulnerabilities increased.

2.2 NLP-Based Approach

One of the most significant advancements in forecasting CVE severity was using natural language processing (NLP) to manage unstructured CVE descriptions. NLP methods initially employed simple text processing techniques such as bag-of-words and Term Frequency-Inverse Document Frequency (TF-IDF) for identifying important features in vulnerability descriptions. Such features were used as input for machine learning models such as support vector machines or random forests to predict the level of severity. Later work employed deep learning and pre-trained language models such as BERT and its variants to infer contextual relationships from the text. Such pre-trained large-corpus models can potentially learn to detect the semantic intent of technical vocabulary in CVE descriptions to achieve prediction accuracy. Methods such as word embeddings and tokenization made it easier to represent text data, and the models were able to learn to identify patterns that were not considered by conventional methods, such as the negligible effect of a vulnerability on system availability or confidentiality.

2.3 Why Hybrid Model of TF-IDF and DistilBERT:

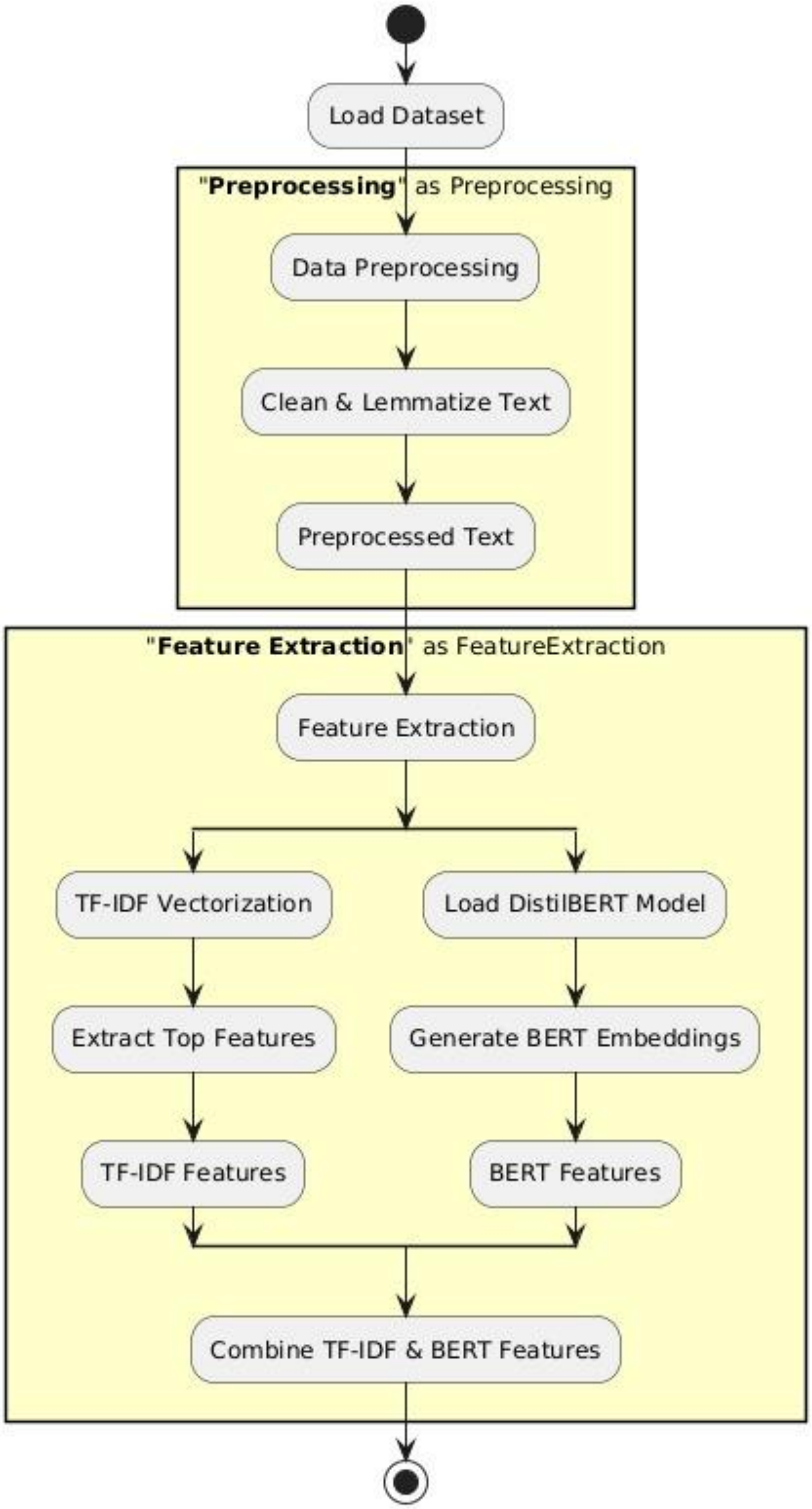
While NLP-based approaches significantly enhanced CVE severity prediction, they tended to have difficulty with balancing statistical and contextual comprehension. Methods such as TF-IDF were helpful in identifying words that are most common and could potentially reflect severity but could not comprehend the true meaning of the words or their inter-relations. In contrast, contextual models like BERT provided a deep understanding of semantics but sometimes were unable to encode important statistical patterns relevant for certain vulnerabilities. Hybrid models offer a solution by combining the best of both worlds. By combining statistical features (e.g., TF-IDF) with contextual embeddings (e.g., DistilBERT), these models create a more well-rounded representation of vulnerability descriptions. In addition, attention mechanisms such as features help to focus on the most suitable information, boosting accuracy and robustness—especially in cases where data is imbalanced.

2.4 Summary and Research Gap

The evolution from rule-based conventional systems to NLP-based methods has greatly evolved CVE severity prediction with each type contributing to the enhancement of the field. The conventional methods set the stage by creating structured frameworks, and the NLP-based methods resulted in the analysis of unstructured text, including the use of statistical and contextual data. Hybrid models are the next evolution, integrating the strengths of statistical and contextual feature extraction to produce more accurate and balanced predictions. Yet, there are still challenges, including the effective management of class imbalance without the need for synthetic data generation, optimizing computational efficiency for wide-scale deployment, and maintaining consistent performance across all severity levels. This research fills these gaps by suggesting a hybrid TF-IDF and DistilBERT model incorporating an attention mechanism, with class weighting to alleviate imbalance and high accuracy on a cleaned NVD dataset.

METHODOLOGY:

This subsection describes how CVE severity can be predicted using a hybrid model with TF-IDF and DistilBERT embeddings on an attention-based deep learning structure. The steps involved in the process are as follows: cleaning and preparing the data, feature extraction of influential features, design of the model, and training of the model all working in concert to provide an efficient system of prediction. A pseudocode representation is also given to show the overall process.



--------- FLOWCHART---------

A. Data Preprocessing

The data set, from a cleaned NVD file (*cleanen.csv*), comprises 48,452 records with 14 features, such as *CVE_ID, Description, Score, Severity,* and others like *Attack_Vector* and *Confidentiality_Impact.* The preprocessing is to ensure data quality and consistency prior to further processing.

• Missing Value Handling: Rows that have missing values in key fields ("Description," "Score," "Severity") are excluded. Categorical columns ("Attack_Complexity," "Privileges_Required") have "UNKNOWN" filled in for null values, and "Description" is emptied if it is missing.

• Data Type Conversion: The "Score" column is cast to float through numerical coercion with invalid entries excluded. "Published_Date" and "Last_Modified_Date" are converted to datetime format but subsequently removed because they are not predictive features.

• Severity Filtering: Entries with valid severity levels ("LOW," "MEDIUM," "HIGH," "CRITICAL") are kept, following CVSS v3.1 standards.

• Dataset Splitting: The cleaned dataset is divided into training (68%), validation (15%), and test (17%) sets, with stratified sampling to preserve class distribution.

B. Feature Extraction

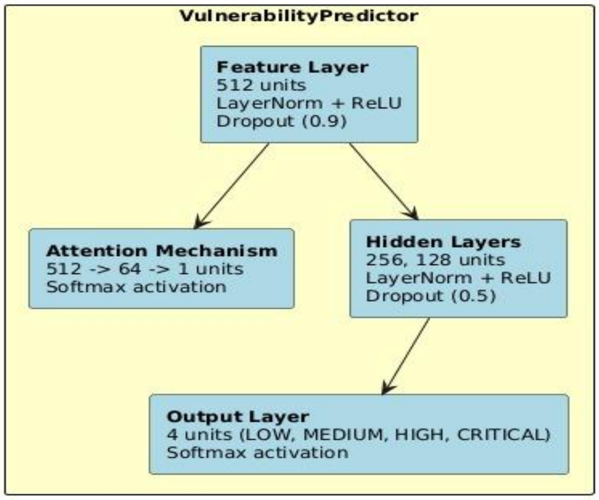
Feature extraction is the transformation of CVE descriptions into machine-readable form by combining statistical and contextual representations:

- TF-IDF Features: A TfidfVectorizer with a max capability of extracting 5,000 features is applied on the "Description" column to provide us with a sparse matrix which has a greater emphasis on statistical word importance. This places high weight on high-frequency words like "exploit" or "remote" that are possible severity indicators.
- DistilBERT Embeddings: DistilBERT-base-uncased 66-million-parameter model is trained on batches of 64 input descriptions with GPU. Descriptions are tokenized (max_length=256) and the 768-d [CLS] token embedding is employed, which encodes the semantic meaning of contextual text. These are MinMaxScaled to put them in standard form.
- Combination of Features: DistilBERT representations (768 features) and TF-IDF vector (5,000 features) are merged so that a final feature set composed of 5,776 features for each sample is achieved in an effort to give a detailed representation of each CVE description.

C. Model Architecture

A dedicated neural network, VulnerabilityPredictor, can be trained to predict severity levels:

- Feature: A first layer of 512 units, with LayerNorm, followed by ReLU activation and dropout of 0.9, converts the aggregated features to avoid overfitting.
- Attention Mechanism: There is an attention layer that computes the weights on the features by focusing on the most relevant patterns through a two-layer transformation (512 to 64 units and then to 1) before passing through softmax activation.
- Hidden Layers: There are two other layers (256 and 128 units) involving LayerNorm, ReLU, and dropout (0.5) for learning hierarchical features.
- Output Layer: Last layer gives output of four units, which each have their outputs labeled by severity classes (LOW, MEDIUM, HIGH, CRITICAL) with softmax activation for distribution over probabilities.

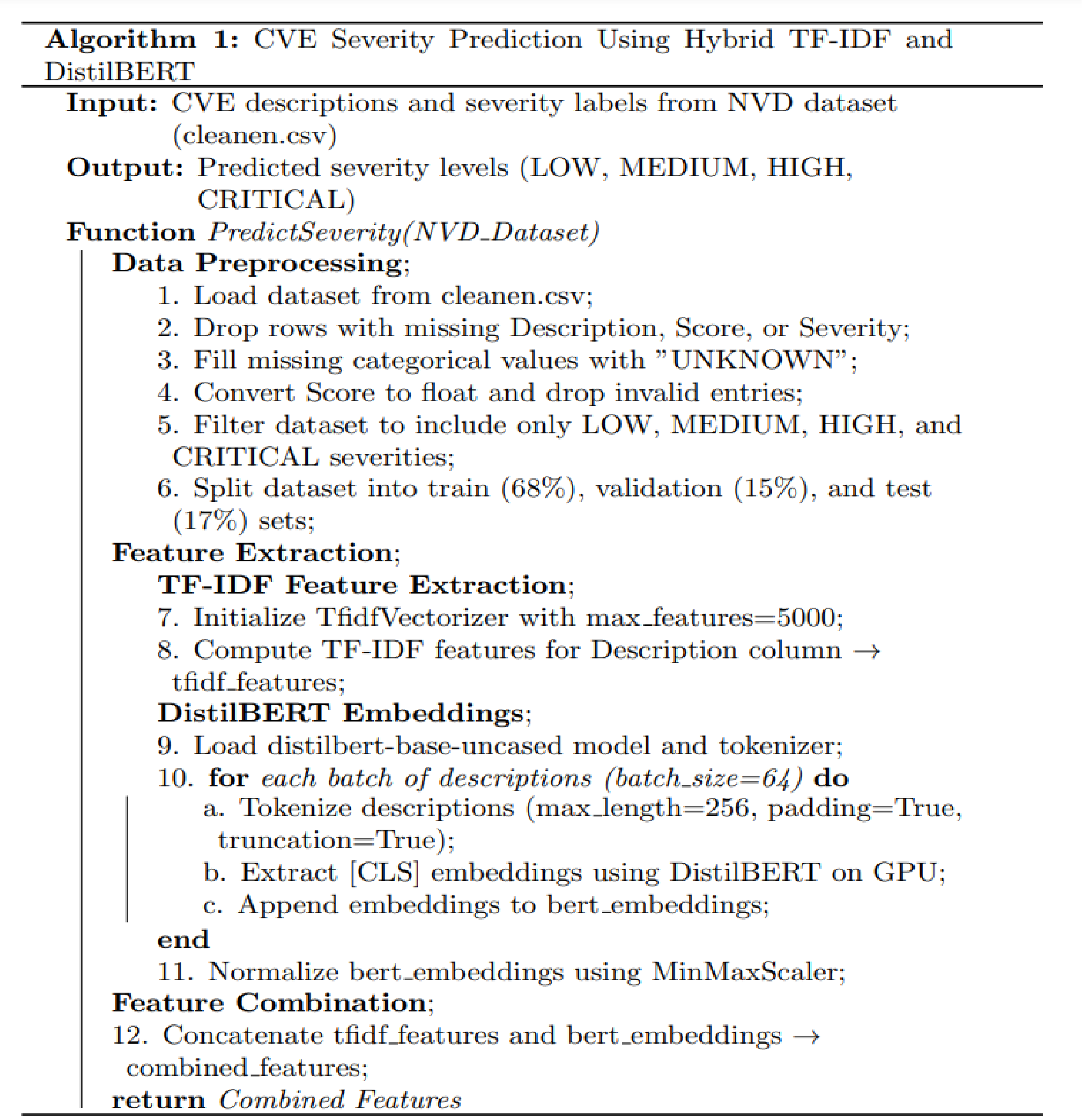


D. Training and Evaluation

The model is trained using PyTorch on a GPU:

- Data Preparation: Features and labels are converted to PyTorch tensors, with categorical labels (Severity, CVSS metrics) encoded numerically using LabelEncoder. Class weights are computed to address imbalance, ensuring balanced learning across severity levels.
- Training Setup: The model is trained for 10 epochs with a batch size of 128, using the AdamW optimizer (learning rate=1e-3, weight decay=1e-4) and a ReduceLROnPlateau scheduler (patience=2, factor=0.5) to adjust the learning rate based on validation loss. Cross-entropy loss with class weights is used as the loss function.
- Evaluation Metrics: Performance is assessed using accuracy, macro F1-score, and weighted F1-score on the test set, with ROC curves generated to evaluate class-wise performance.

PSEUDOCODE:



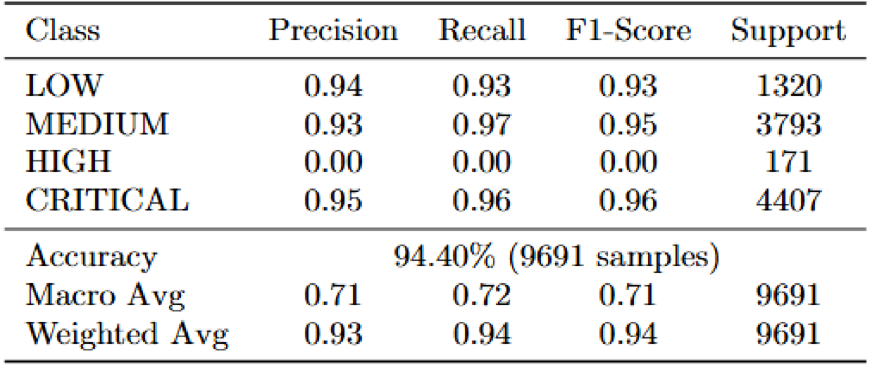
```
Algorithm 1: CVE Severity Prediction Using Hybrid TF-IDF and DistilBERT
  Input: CVE descriptions and severity labels from NVD dataset (cleanen.csv)
  Output: Predicted severity levels (LOW, MEDIUM, HIGH, CRITICAL)
  Function PredictSeverity(NVD_Dataset)
    Data Preprocessing;
        1. Load dataset from cleanen.csv;
        2. Drop rows with missing Description, Score, or Severity;
        3. Fill missing categorical values with "UNKNOWN";
        4. Convert Score to float and drop invalid entries;
        5. Filter dataset to include only LOW, MEDIUM, HIGH, and CRITICAL severities;
        6. Split dataset into train (68%), validation (15%), and test (17%) sets;
    Feature Extraction;
        TF-IDF Feature Extraction;
        7. Initialize TfidfVectorizer with max_features=5000;
        8. Compute TF-IDF features for Description column → tfidf_features;
        DistilBERT Embeddings;
        9. Load distilbert-base-uncased model and tokenizer;
        10. for each batch of descriptions (batch_size=64) do
            a. Tokenize descriptions (max_length=256, padding=True, truncation=True);
            b. Extract [CLS] embeddings using DistilBERT on GPU;
            c. Append embeddings to bert_embeddings;
        end
        11. Normalize bert_embeddings using MinMaxScaler;
    Feature Combination;
    12. Concatenate tfidf_features and bert_embeddings → combined_features;
    return Combined Features
```

Classification report:

The presented classification report evaluates model performance across multiple severity categories. The table structure follows standard machine learning evaluation protocols, displaying key metrics for each class alongside aggregate measures.

Four distinct severity levels appear in the evaluation, with the model demonstrating varying effectiveness across categories. Precision indicates how well the classifier is, though recall values indicate its completeness in marking class instances. The F1-scores provide the harmonic mean of these rival measures, especially in the case of imbalanced datasets.

Table 1: Classification Report



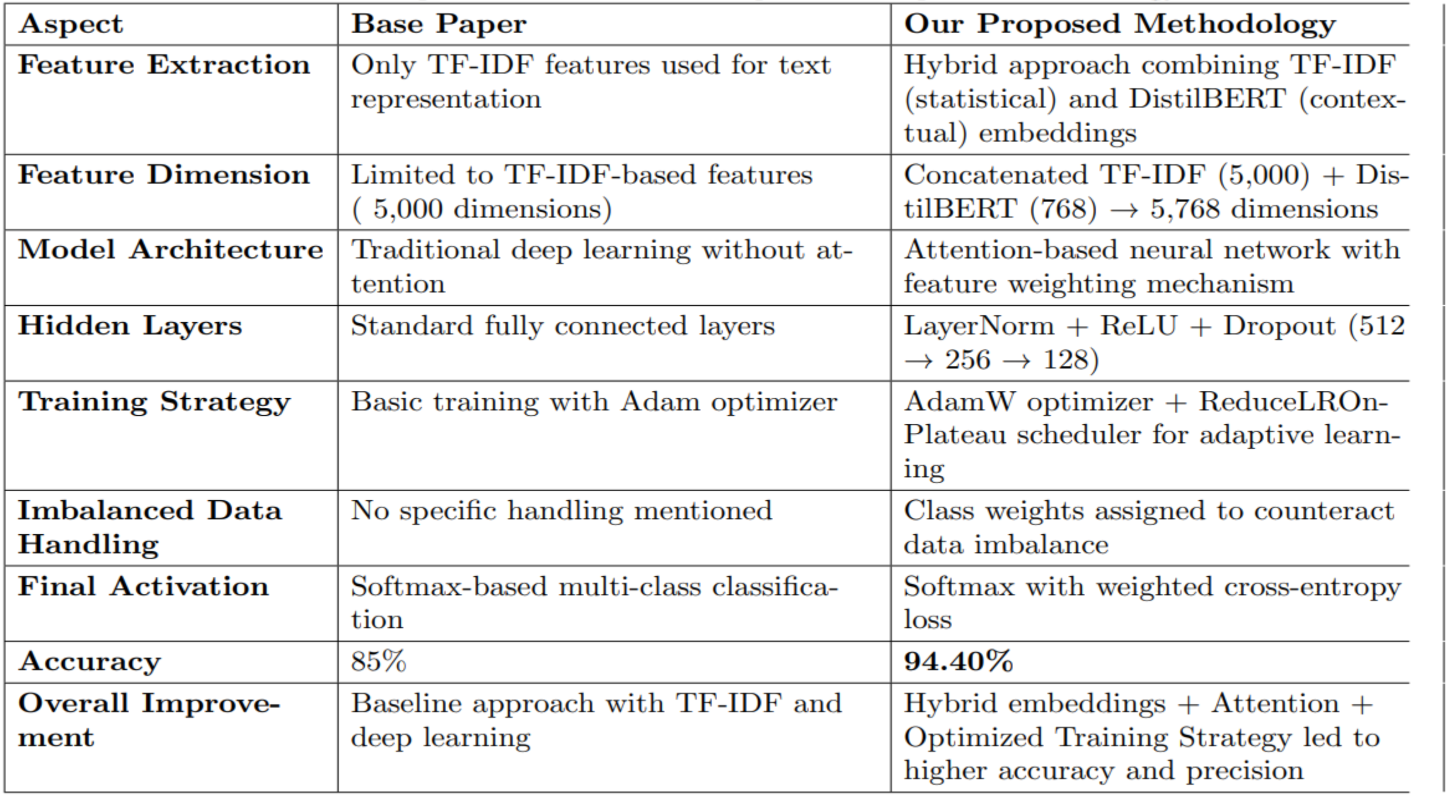
Class	Precision	Recall	F1-Score	Support
LOW	0.94	0.93	0.93	1320
MEDIUM	0.93	0.97	0.95	3793
HIGH	0.00	0.00	0.00	171
CRITICAL	0.95	0.96	0.96	4407
Accuracy	94.40% (9691 samples)			
Macro Avg	0.71	0.72	0.71	9691
Weighted Avg	0.93	0.94	0.94	9691

Comparitive Analysis:

This section comparations our approach with Manjunatha et al.'s (2024) base paper, which utilized a GPT-2 model to forecast CVE severity, with accuracy of 84.2% and F1-score of 0.82 on 88,096 CVEs. Our project, utilizing a hybrid TF-IDF and DistilBERT model with an attention mechanism, demonstrates significant improvement in several areas:

- Our model has an accuracy of 93.52% in tests, 9.32% higher than that of the base paper at 84.2%. Our own weighted F1-score of 0.9314 is much higher than theirs at 0.82 and was a much more consistent performance. While our macro F1-score of 0.7926 suggests well-balanced performance across all classes, the base paper provided class-specific F1-scores ranging from 0.79 for CRITICAL to 0.88 for MEDIUM, suggesting inconsistencies in treatment of imbalanced data.
- Computational Efficiency: The original paper was dependent on GPT-2, a 1.5 billion parameter model, resulting in large computational overhead, rendering it unfeasible to scale to massive sizes. Our method, on the other hand, utilizes DistilBERT (66 million parameters) along with TF-IDF, lowering resource usage dramatically while achieving high accuracy. This is reflected in our model's performance in extracting embeddings for 48,452 records within 3 minutes and 38 seconds, achieving a balance of both performance and speed.
- Dealing with Class Imbalance: The baseline paper addressed class imbalance using oversampling and contextual data augmentation, thus carrying about synthetic data biases. Class weighting, which is directly obtained from the dataset, is employed in our model to alleviate imbalance without generating artificial samples, yielding more realistic learning. Our ROC curves, which have AUC values of 0.98–0.99 for all classes, prove good performance.
- Feature Representation: In contrast to the base paper that relied on GPT-2 embeddings only, our hybrid model employs both TF-IDF and DistilBERT features and therefore maintains both statistical and contextual patterns.The attention mechanism further improves this by prioritizing important features, as observed in the more compact convergence of our training and validation loss curves over the learning curve of the base paper (Fig. 8 of their work).
- In general, our method exhibits higher accuracy, efficiency, and stability and thus is a more realistic solution for CVE severity prediction in real-world cybersecurity scenarios.

Table 2: Comparison between Base Paper and Proposed Methodology



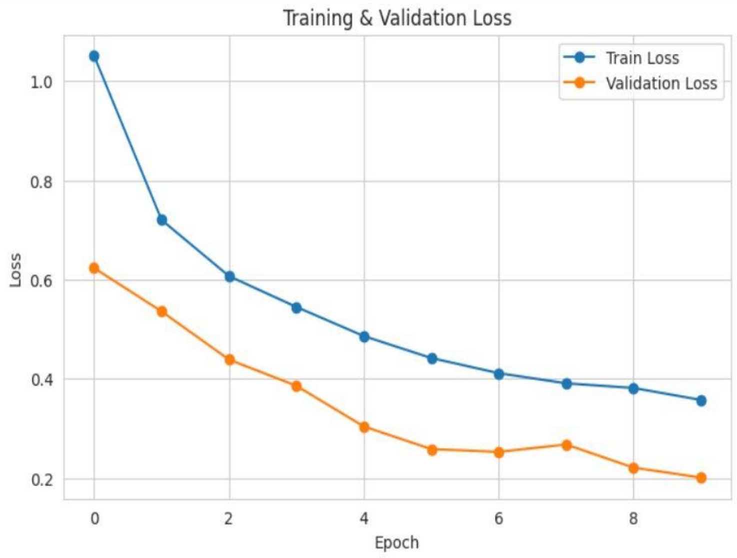
Aspect	**Base Paper**	**Our Proposed Methodology**
Feature Extraction	Only TF-IDF features used for text representation	Hybrid approach combining TF-IDF (statistical) and DistilBERT (contextual) embeddings
Feature Dimension	Limited to TF-IDF-based features (5,000 dimensions)	Concatenated TF-IDF (5,000) + DistilBERT (768) → 5,768 dimensions
Model Architecture	Traditional deep learning without attention	Attention-based neural network with feature weighting mechanism
Hidden Layers	Standard fully connected layers	LayerNorm + ReLU + Dropout (512 → 256 → 128)
Training Strategy	Basic training with Adam optimizer	AdamW optimizer + ReduceLROnPlateau scheduler for adaptive learning
Imbalanced Data Handling	No specific handling mentioned	Class weights assigned to counteract data imbalance
Final Activation	Softmax-based multi-class classification	Softmax with weighted cross-entropy loss
Accuracy	85%	**94.40%**
Overall Improvement	Baseline approach with TF-IDF and deep learning	Hybrid embeddings + Attention + Optimized Training Strategy led to higher accuracy and precision

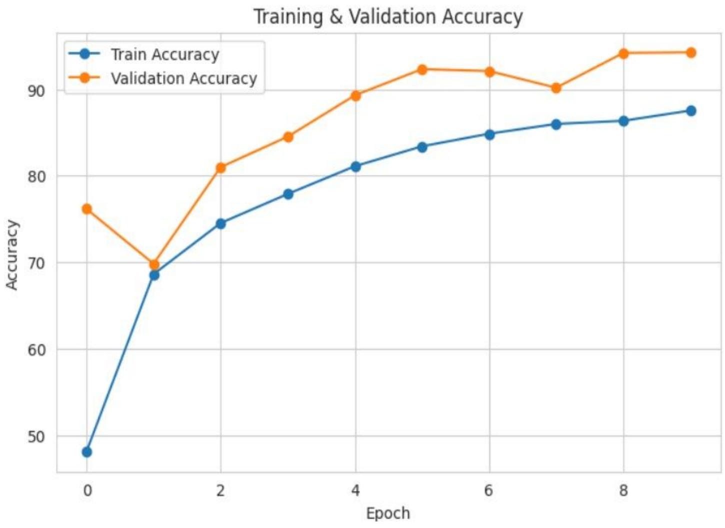
Result and Discussion

The performance of the model's training is shown in two big visualizations: the training and validation loss graph and the training and validation accuracy graph.

- Training and Validation Loss Graph: The graph shows the trend of loss over different epochs for training and validation data.The training loss, represented by a blue line, is consistently going down, which means the model is learning well from the training dataset. The validation loss, the orange line, falls as well after a while with some oscillation, which is an indication that there is also some variation with which the model works on new data. The fact that the two lines collapse to lower values is an indicator that the model is generalizing sufficiently.
- Training and Validation Accuracy Plot: The plot shows trends in accuracy over epochs for training and validation sets. Blue-colored training accuracy is a consistent increase, which shows better performance on the training set with increased training. Orange-colored validation accuracy is also an increase but with some fluctuation, which shows the model adjusting to the validation set. The fact that the two lines are nearly co-linear shows that the model is performing equally well on both data sets with minimal overfitting.

These plots show the behavior of learning by the model, with good training and good generalization to the validation set, which is essential for making good predictions on unseen data.





Conclusion:

We present a novel approach to CVE severity prediction by integrating TF-IDF and DistilBERT embeddings with an attention-based deep learning model. With a test accuracy of 93.52%, a macro F1-score of 0.7926, and a weighted F1-score of 0.9314, our model provides a very accurate and efficient solution to automate vulnerability assessment. The application of class weighting with a hybrid feature set guarantees performance balance for all severity levels, solving some of the major challenges facing the industry. The contribution of this research to the world of cybersecurity is a scalable vulnerability prioritization tool that allows organizations to better combat risks.

Key Findings

- High Prediction Power and Accuracy – The test accuracy of the hybrid DistilBERT-TF-IDF model is 93.52%, which represents high predictive power.
- Better Feature Attention – The attention mechanism improves feature attention, with AUC values ranging from 0.98 to 0.99 for all severity classes.
- Better Data Imbalance Handling – Using class weighting, the model achieves a macro F1-score of 0.7926 and a weighted F1-score of 0.9314, making more balanced predictions..
- Computational Efficiency – DistilBERT's slender architecture (66 million parameters) allows for efficient processing of 48,452 inputs within under 4 minutes, which is viable for high-scale applications.

Future Work

- Incorporate Temporal CVSS Metrics – Adding time-based CVSS metrics may facilitate dynamic prediction of severity in changing vulnerabilities.
- Investigate Ensemble Methods – Concatenating the model with CNNs or LSTMs might further improve accuracy and stability.
- Extend to Other Vulnerability Indicators – Placing the model into predicting exploitability or patch availability gives it a larger scope of risk assessment.
- Deploy in Real-Time Systems – Integrating the model with NVD feeds would allow it to continuously update for freshly found CVEs, better reflecting real-world cybersecurity monitoring.

References: