

Article

IoT Vulnerability Severity Prediction Using Lightweight Transformer Models

Samira A. Baho and Jemal Abawajy * 

Faculty of Science, Engineering and Built Environment, Deakin University, Waurin Ponds Campus, Locked Bag 20000, Geelong, VIC 3220, Australia; sbaho@deakin.edu.au

* Correspondence: jemal@deakin.edu.au

Abstract

Vulnerability severity assessment plays a critical role in cybersecurity risk management by quantifying risk based on vulnerability disclosure reports. However, interpreting these reports and assigning reliable risk levels remains challenging in Internet of Things (IoT) environments. This paper proposes an IoT vulnerability severity prediction framework aligned with the Common Vulnerability Scoring System (CVSS). The framework is based on a lightweight transformer architecture. It uses a distilled version of Bidirectional Encoder Representations from Transformers (BERT). The model is fine-tuned using transfer learning to capture contextual semantic information from vulnerability descriptions. The lightweight design preserves computational efficiency. Experimental evaluation on an IoT vulnerability dataset shows strong and consistent performance across all severity classes. The proposed model achieves double-digit improvements across key evaluation metrics. In most cases, the improvement exceeds 20% compared with traditional machine learning and baseline deep learning approaches. These results show that lightweight transformer models are well suited for IoT security. They provide a practical and effective solution for automated vulnerability severity classification in resource- and data-constrained environments.

Keywords: IoT security; vulnerability severity prediction; natural language processing; transformer models; DistilBERT; transfer learning; cybersecurity risk assessment

1. Introduction

IoT has become an integral component of modern digital ecosystems. It enables connectivity across a wide range of applications, including smart homes, healthcare, industrial automation, transportation, and critical infrastructure. The rapid proliferation of IoT devices has significantly expanded the cyber-attack surface. As a result, IoT environments are increasingly exposed to security threats and vulnerabilities [1–3].

To contextualise these security challenges and the expanded attack surface, Figure 1 presents a three-layer IoT security architecture [4]. The architecture categorises attack surfaces across the device, network, and application layers. Layer 1, the IoT Device or Perception Layer, comprises physical hardware such as CCTV cameras, wearables, and medical sensors. Vulnerabilities at this layer include hardcoded credentials and insecure firmware, which can provide attackers with an initial foothold. Layer 2, the Network or Communication Layer, supports data transmission through protocols such as Wi-Fi and Bluetooth. This layer is vulnerable to weaknesses such as weak pre-shared keys and outdated protocols, which may lead to traffic interception or unauthorised network access. Layer 3, the Application Layer, includes services such as mobile applications, analytics or



Academic Editors: Kamran Siddique,
Ka Lok Man and Zahid Akhtar

Received: 26 December 2025

Revised: 5 February 2026

Accepted: 9 February 2026

Published: 14 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

artificial intelligence engines, and cloud databases. Security risks at this layer primarily involve unauthorised access and poor input validation, which may result in data breaches or malicious command injection. This structured view motivates the proposed automated vulnerability analysis pipeline.

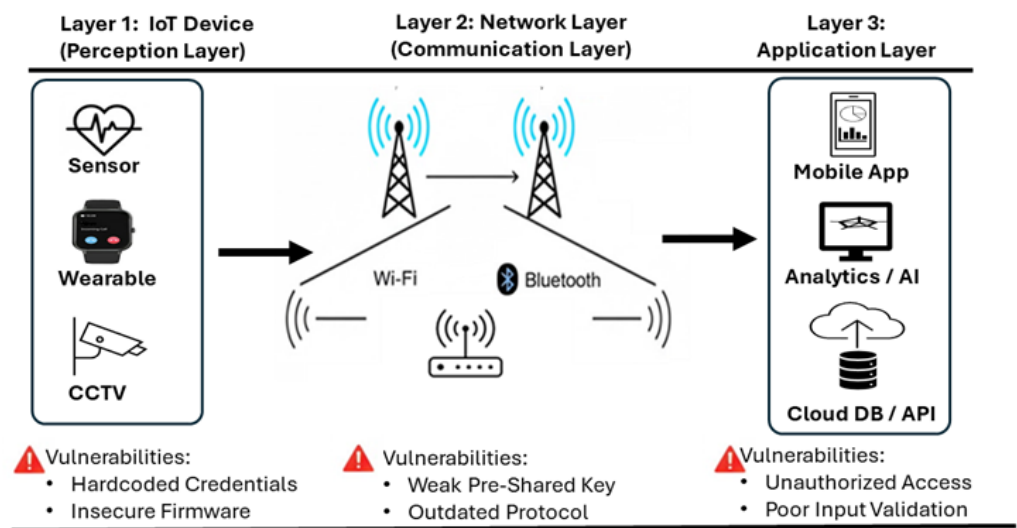


Figure 1. Overview of a typical IoT ecosystem and common security vulnerabilities across device, network, and cloud layers.

This layered perspective highlights the diverse and pervasive nature of security weaknesses across IoT systems. Effective vulnerability management is therefore essential for maintaining system security, service availability, and safety in IoT environments [4,5]. A key component of this process is vulnerability severity assessment. It quantifies threat impact and determines remediation urgency. Severity information enables security teams to prioritise patching efforts and allocate limited resources. In practice, vulnerability severity is commonly assessed using CVSS [6], an expert-driven framework based on exploitability and impact metrics. However, prior studies identify several limitations, including subjectivity, limited contextual awareness, and poor scalability as vulnerability disclosures continue to increase [6]. To address these limitations, recent research has explored automated vulnerability severity prediction using machine learning (ML), deep learning (DL), and natural language (NLP) processing techniques. Early approaches relied on traditional ML models applied to handcrafted textual features extracted from vulnerability descriptions [7]. More recent studies demonstrate that deep learning models can learn severity-relevant patterns directly from unstructured vulnerability text, improving prediction accuracy and consistency [8,9]. Transformer-based language models further advance severity prediction by enabling richer contextual understanding of vulnerability descriptions. In particular, BERT-based models have shown strong performance in predicting CVSS metrics and severity levels directly from text [9–13].

Despite these advances, most existing approaches rely on IT-centric vulnerability datasets such as the National Vulnerability Database (NVD). The applicability of these prior approaches to IoT-specific vulnerabilities remains insufficiently explored [5,14]. IoT vulnerabilities differ from conventional software vulnerabilities due to heterogeneous device architectures, constrained resources, diverse deployment contexts, and long device lifecycles. These characteristics are not well captured by IT-focused datasets, which limits the generalisation of existing severity prediction models to IoT environments [4]. In addition, the limited availability of well-annotated IoT-specific vulnerability data poses challenges for training deep learning models under data-scarce conditions. Although efforts

such as the VARIOt repository provide IoT-focused datasets, empirical studies leveraging these resources for automated severity prediction remain limited [15]. Consequently, there is a lack of systematic evidence on how modern transformer-based models can be effectively adapted for IoT vulnerability severity prediction.

To address this gap, this paper proposes an automated approach for predicting IoT vulnerability severity from textual descriptions using a fine-tuned lightweight transformer-based model. To the best of our knowledge, no prior work has systematically investigated lightweight Transformer-based severity prediction using IoT-specific vulnerability descriptions under explicitly data-scarce conditions. In this work, the term lightweight refers to architectural and parameter efficiency relative to standard transformer models, enabling operation under practical IoT constraints such as class imbalance and limited computational resources. The approach predicts CVSS-derived categorical severity levels by classifying vulnerabilities as Low, Medium, High, or Critical. These predictions are not intended to replace the full CVSS scoring process, but rather to provide an approximate severity indication derived from textual descriptions.

The main contributions of this paper are summarised as follows:

- We propose a transformer-based approach for IoT vulnerability severity prediction that leverages transfer learning to adapt a lightweight pre-trained language model to IoT-focused vulnerability data.
- We empirically evaluate the effectiveness of the proposed model for IoT vulnerability severity prediction using an IoT-specific dataset, unlike prior transformer-based studies that primarily focus on IT-centric data.
- We conduct a comprehensive evaluation against traditional machine learning models and deep learning baselines, demonstrating that the proposed approach achieves high accuracy and robust performance despite limited training data.
- We show that lightweight transformer models can provide an effective balance between predictive performance and computational efficiency for automated vulnerability severity assessment in resource-constrained IoT environments.

The rest of this paper is organised as follows. Section 2 presents background and problem formulation. Section 3 describes the proposed methodology. Section 4 reports the experimental evaluation. Section 5 presents the results, followed by discussion in Section 6. Section 7 concludes the paper and outlines directions for future work.

2. Related Work

In this section, we review recent research and developments in vulnerability severity quantification, a field that has seen a significant surge in scholarly attention.

2.1. CVSS-Based Method

Vulnerability severity is commonly assessed using CVSS [6]. CVSS v3.1 is the most widely used version for computing overall vulnerability severity through the Base Score, which ranges from 0.0 to 10.0. The Base Score is derived from the Exploitability and Impact metric groups. Table 1 summarises the metrics and categorical values used in this calculation. However, previous research highlights several limitations of CVSS, including subjectivity, susceptibility to human error, limited contextual awareness, and poor scalability as vulnerability disclosures continue to increase [6].

CVSS is an expert-driven framework based on exploitability and impact metrics. As the volume of reported vulnerabilities continues to increase, automated vulnerability severity prediction has attracted growing interest from the research community as a scalable alternative to manual assessment. In particular, automated severity prediction from textual vulnerability descriptions has gained attention as manual CVSS-based evaluation becomes

increasingly time-consuming, subjective, and difficult to scale [6,16]. This automation is critical due to the sheer volume of new vulnerabilities reported daily, which often outpaces the capacity of human analysts.

Table 1. CVSS v3.1 Base Metrics and Values.

Metric Group	Base Metric	Possible Values (Classes)
Exploitability	Attack Vector (AV)	Network (N), Adjacent (A), Local (L), Physical (P)
	Attack Complexity (AC)	Low (L), High (H)
	Privileges Required (PR)	None (N), Low (L), High (H)
Scope	User Interaction (UI)	None (N), Required (R)
	Scope (S)	Unchanged (U), Changed (C)
Impact	Confidentiality (C)	None (N), Low (L), High (H)
	Integrity (I)	None (N), Low (L), High (H)
	Availability (A)	None (N), Low (L), High (H)

2.2. ML-Based Methods

Early approaches to automated vulnerability severity prediction primarily relied on ML techniques applied to handcrafted textual features extracted from vulnerability descriptions [7,17]. Malhotra and Vidushi employ traditional machine learning classifiers with handcrafted textual features to predict software vulnerability severity from textual descriptions [7]. Similarly, Jiang and Atif [17] propose a rule-based and traditional machine learning approach for automated vulnerability discovery and severity assessment in cyber-physical systems. These methods demonstrated initial feasibility. However, their performance was constrained by limited feature engineering and insufficient semantic understanding of complex vulnerability narratives.

2.3. DL-Based Methods

To overcome ML limitations, researchers have increasingly adopted DL-based models for vulnerability severity prediction [8,18,19]. For example, convolutional neural network (CNN)-based approaches have been widely explored due to their ability to learn hierarchical textual features. Saklani and Kalia demonstrated that CNN-based models outperform classical ML classifiers for vulnerability severity prediction using textual data [8]. Wang et al. proposed a deep learning model called Latent Space Networks (LSNets), an adaptive latent space network designed to model complex feature interactions for vulnerability severity assessment [19]. More advanced learning paradigms have also been explored to capture complex relationships within vulnerability data. Shan et al. introduced a multi-task deep learning framework that jointly predicts multiple vulnerability-related attributes, improving generalisation through shared representations [18]. They compare their multi-task deep learning model against traditional machine learning baselines, demonstrating improved performance in vulnerability severity prediction. While DL-based approaches improve prediction accuracy over traditional machine learning models, they often suffer from limited contextual understanding, high data requirements, and reduced suitability for resource-constrained environments.

2.4. Transformer-Based Methods

Transformer-based language models have further advanced automated vulnerability severity and CVSS prediction, particularly in general IT and software vulnerability contexts. Several transformer-based methods focusing on automated CVSS score prediction from CVE textual descriptions have been proposed. Shahid and Debar introduced CVSS-BERT, which employs multiple BERT-based classifiers, each dedicated to predicting an individual

CVSS metric [10]. By fine-tuning each classifier independently, the approach achieves high accuracy for individual metric prediction and produces severity scores that closely align with expert assessments. To reduce computational overhead, the authors adopt BERT-small, a reduced BERT variant with fewer layers and parameters. While CVSS-BERT is trained on 2018–2020 general NVD vulnerability descriptions to predict specific CVSS v3.1 base metrics, our work focuses on IoT-specific descriptions to predict qualitative severity levels, ranging from Low to Catastrophic.

Costa et al. investigated DistilBERT for predicting individual CVSS metrics directly from vulnerability descriptions [11], demonstrating that transformer-based models can approximate expert scoring using semantic information alone. DistilBERT, a distilled version of BERT, was introduced to reduce model size and inference cost while preserving much of BERT's representational power [13]. Zhang et al. proposed an AI-enabled framework for automated CVSS scoring using contextual embeddings, further reinforcing the effectiveness of transformer-based models for vulnerability severity estimation [20]. Overall, these studies report strong alignment between predicted CVSS scores and expert-provided assessments.

While several studies have explored lightweight transformer variants such as BERT-small and DistilBERT to improve computational efficiency, these approaches are primarily evaluated on conventional software or IT vulnerability datasets (e.g., NVD-style CVE descriptions). Their primary focus is on CVSS metric or vector prediction, rather than direct severity level classification. Moreover, they do not explicitly address challenges specific to IoT-focused vulnerability datasets, including limited labeled data, heterogeneous device descriptions, contextual variability, and constrained deployment environments [5].

In contrast, this work systematically focuses on severity level prediction using IoT-specific vulnerability descriptions under data-scarce conditions, thereby addressing a gap not covered by existing transformer-based approaches.

2.5. Hybrid Methods

Hybrid models typically combine transformer-based language models with additional neural architectures (e.g., BERT-CNN, BERT-LSTM) or with rule-based and handcrafted feature pipelines, primarily in general IT and software vulnerability analysis contexts. Ni et al. proposed a BERT-CNN hybrid model that integrates contextual embeddings from BERT with CNN-based feature extraction, achieving improved performance over standalone architectures [9]. Building on this paradigm, Mirtaheri et al. demonstrated that fine-tuned BERT-CNN architectures can further enhance automated vulnerability scoring, highlighting the benefits of combining contextual language representations with deep feature extractors [21].

Marali and Balakrishnan applied fine-tuned BERT models combined with deep neural networks to vulnerability classification, achieving improved accuracy compared to traditional approaches [12]. Aghaei et al. investigated automated CVE analysis for threat prioritisation and impact prediction, proposing a hybrid transformer-based approach that combines SecureBERT, built on RoBERTa, with TF-IDF features for automated CVSS vector prediction [22,23].

More recently, Mirtaheri et al. proposed a hybrid generative AI and transformer-based framework for automated vulnerability scoring, combining a large language model (e.g., GPT-3.5-Turbo) with a fine-tuned BERT-small model to balance accuracy and computational cost. While effective in data-rich and cloud-based settings, such hybrid and LLM-assisted approaches assume substantial computational resources and are primarily evaluated on conventional IT vulnerability datasets, limiting their applicability to resource-constrained and data-scarce IoT environments.

2.6. Summary and Research Gap

IoT security has received increasing attention due to the rapid growth of connected devices and their heightened exposure to cyber threats, particularly in heterogeneous and resource-constrained environments [2,4]. Despite progress in automated vulnerability severity prediction, most existing studies rely heavily on CVE data from the National Vulnerability Database and primarily target conventional IT and enterprise software vulnerabilities. Prior work has highlighted fundamental structural and contextual differences between IoT and traditional IT systems, underscoring the need for IoT-specific vulnerability analysis frameworks that better reflect device heterogeneity, deployment context, and operational constraints [4,14].

IoT-focused repositories such as VARIOt [15] differ substantially from IT-centric datasets in reporting practices, contextual emphasis, and metadata availability. As a result, they place greater reliance on unstructured textual descriptions and limit the direct applicability of models trained on conventional CVE data [4,15]. Furthermore, the scarcity of well-annotated IoT-specific datasets constrains the effective training of large deep learning models from scratch. While much prior work focuses on predicting individual CVSS metrics or vectors, comparatively little attention has been given to directly predicting CVSS-derived categorical severity levels for IoT vulnerabilities.

This study addresses these limitations by systematically focusing on vulnerability severity level classification using IoT-specific textual descriptions under data-scarce and CPU-only conditions that are common in IoT security deployments. Such settings impose constraints including limited labelled data, class imbalance, and restricted computational resources, particularly in edge or on-premise environments. Consequently, this work avoids training models from scratch and the use of computationally intensive architectures or hybrid pipelines that rely on large-scale models or cloud-based inference.

Instead, the proposed approach employs DistilBERT [13], a lightweight transformer that leverages transfer learning to adapt pre-trained language representations to IoT-focused vulnerability data. DistilBERT preserves much of BERT's semantic and predictive capability while reducing the number of parameters by approximately 40% and significantly lowering inference cost [13,24]. This balance between efficiency and effectiveness makes DistilBERT well suited for IoT vulnerability severity assessment under data-scarce and resource-constrained conditions.

3. IoT Vulnerability Severity Prediction Model

This section presents the proposed framework for IoT vulnerability severity prediction based on textual descriptions. We refer to the proposed framework as IoT Distilled Bidirectional Encoder Representations from Transformers (IoTDistilBERT). The objective of the framework is to accurately classify IoT vulnerabilities into predefined severity levels by leveraging semantic information embedded in unstructured vulnerability reports, while addressing challenges related to data scarcity, contextual variability, and computational efficiency.

3.1. Problem Formulation

Vulnerability severity prediction in IoT environments can be formulated as a supervised multi-class text classification task. In this setting, unstructured vulnerability descriptions contain latent semantic information related to exploitability, impact, attack complexity, and deployment context. The objective is to infer this information and map each vulnerability description to an appropriate severity level that reflects its associated security risk. Advances in natural language processing and representation learning have shown that such semantic information can be effectively captured using distributed representations

learned by deep neural architectures, providing a principled foundation for automated, data-driven vulnerability severity assessment. In this study, the focus is on predicting CVSS-derived categorical severity levels rather than estimating continuous CVSS base scores. Formally, let

$$\mathcal{D} = \{(d_i, y_i)\}_{i=1}^N \quad (1)$$

denote a labelled dataset of IoT vulnerability reports, where each d_i represents an unstructured textual description and y_i denotes the corresponding severity label. The label space is defined as

$$\mathcal{Y} = \{Low, Medium, High, Critical\} \quad (2)$$

Each vulnerability description d_i may include information about affected devices or components, attack vectors, exploitation conditions, and potential consequences. The objective is to learn a classification function:

$$f_{\theta} : \mathcal{D} \rightarrow \mathcal{Y} \quad (3)$$

parameterised by θ , that maps a vulnerability description to its correct severity class. The function f_{θ} is learned by minimising a suitable multi-class classification loss over the training data, enabling the model to capture semantic patterns that distinguish severity levels based on textual evidence.

This formulation enables direct inference of CVSS-aligned severity categories from textual descriptions, supporting automated, scalable, and consistent severity prediction. Such an approach is particularly well suited for IoT environments characterised by rapidly increasing vulnerability volumes, heterogeneous devices, and limited expert resources. The choice of a categorical multi-class formulation was motivated by both practical and data-driven considerations. First, categorical severity levels are commonly used in operational vulnerability management and provide a familiar abstraction for summarising vulnerability impact. This contextual relevance motivates their use as prediction targets, without implying a specific downstream decision-making objective. Second, CVSS scores are derived from multiple expert-assigned metrics and are subject to annotation noise and subjectivity, particularly in IoT vulnerability reports. Predicting exact numerical scores under data-scarce conditions can amplify this noise, whereas categorical labels provide a more robust and stable target. Third, available IoT-specific datasets exhibit class imbalance and limited sample sizes, making regression and ordinal learning approaches less stable and more sensitive to outliers. Under these constraints, multi-class classification offers a more reliable formulation. Finally, although severity categories exhibit an inherent ordinal relationship, this study does not explicitly exploit ordinal structure. The primary objective is to evaluate whether lightweight transformer models can reliably distinguish severity levels under realistic IoT deployment constraints. Exploring ordinal or regression-based formulations is left for future work as larger and more consistently annotated IoT datasets become available.

To formulate the multi-class classification task, CVSS base scores were mapped to four categorical severity levels (Low, Medium, High, and Critical) following standard CVSS v3.x guidelines [25]. Numerical CVSS scores were discretised using the official threshold ranges: Low (0.1–3.9), Medium (4.0–6.9), High (7.0–8.9), and Critical (9.0–10.0). Scores lying exactly on category boundaries were consistently assigned to the higher severity class to ensure unambiguous labeling. Table 2 presents the mapping between CVSS base scores and categorical severity labels.

Records with missing, incomplete, or undefined CVSS base scores were excluded from the labeling process, as such cases do not permit reliable severity categorisation.

This filtering step ensured that all instances used for model training and evaluation were associated with well-defined and comparable ground-truth labels.

Table 2. Mapping of CVSS base scores to categorical severity labels.

CVSS Base Score Range	Severity LABEL			
	Low 0.1–3.9	Medium 4.0–6.9	High 7.0–8.9	Critical 9.0–10.0

3.2. Framework Overview

The proposed framework follows a systematic Natural Language Processing (NLP) pipeline designed to transform raw IoT vulnerability descriptions into accurate severity predictions. The end-to-end architecture, illustrated in Figure 2, adopts a transformer-based approach to map unstructured textual inputs to discrete vulnerability severity classes. The pipeline consists of four main stages: data acquisition, text preprocessing and tokenization, contextual representation learning, and severity classification. In the data acquisition stage, raw IoT vulnerability descriptions are collected from an IoT-focused vulnerability repository. These descriptions typically contain technical narratives describing exploitation mechanisms, affected devices or firmware, attack vectors, and potential system impact. As such, they serve as the primary source of semantic information required for severity inference.

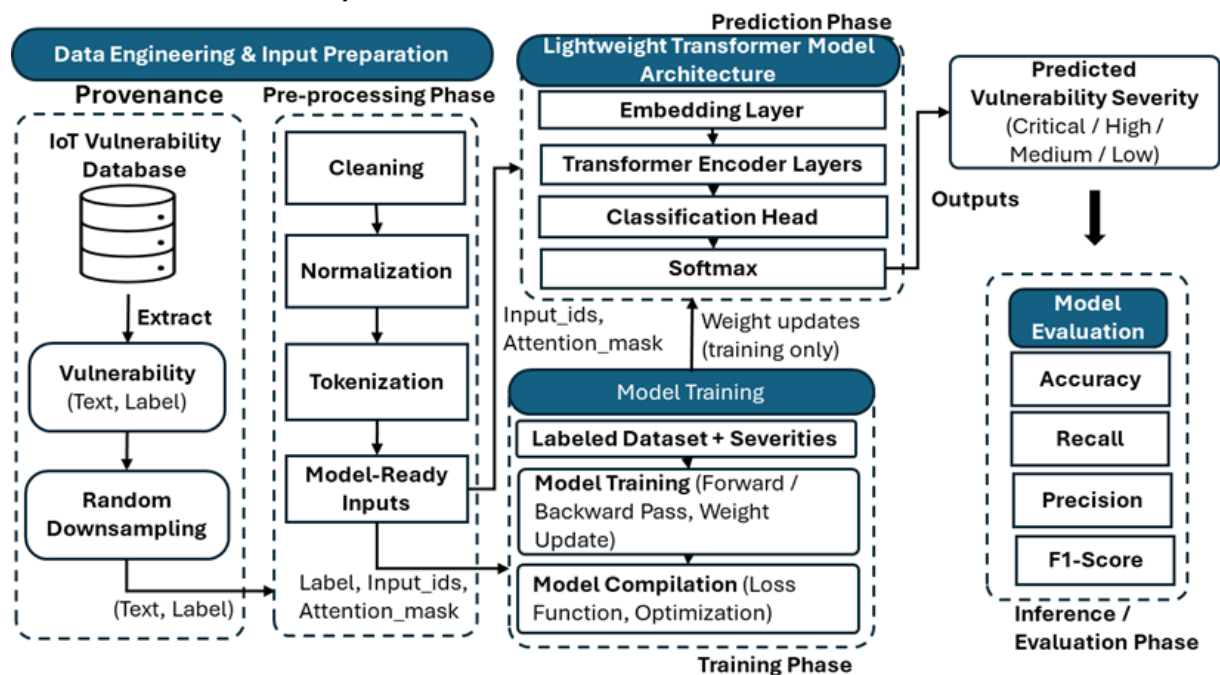


Figure 2. IoT Vulnerability Severity Prediction Framework.

Figure 2 illustrates the end-to-end pipeline for IoT vulnerability severity prediction using a fine-tuned DistilBERT model, encompassing data preparation, model architecture, training, and evaluation. The process begins with data engineering and input preparation. Vulnerability records, consisting of textual descriptions and associated severity labels, are extracted from an IoT vulnerability database and subjected to random downsampling to mitigate class imbalance.

The textual data then undergoes a preprocessing pipeline that includes text cleaning, basic text normalization, tokenization, and input tensor creation, resulting in model-ready inputs. During this stage, noise, special characters, and irrelevant symbols are removed

from the text. Text normalization is limited to basic formatting operations such as lowercasing and whitespace normalization and does not include stopword removal, stemming, or lemmatization. The processed text is subsequently segmented into subword tokens using a transformer-compatible tokenizer.

Vulnerability descriptions are transformed into model-ready inputs (`input_ids` and `attention_mask`) during preprocessing and are used by the model during both training and inference. The categorical severity labels are encoded as integers and used exclusively during training for loss computation and weight updates. The `input_ids` are integer indices of tokens derived from the DistilBERT tokenizer vocabulary and include special tokens (e.g., [CLS], [SEP]). Attention masks specify which tokens should be attended to by the transformer and which should be ignored, such as padding tokens. These fixed-length input representations are suitable for DistilBERT-based classification. They enable efficient processing of long, information-dense vulnerability descriptions and support batch training under CPU-only and resource-constrained settings.

The prepared inputs are then fed into the fine-tuned DistilBERT model architecture, which comprises an embedding layer, multiple Transformer encoder layers, and a task-specific classification head followed by a softmax layer. Within the model, token IDs are transformed into dense contextual embeddings, which are processed by the encoder layers and classification head to produce vulnerability severity predictions.

Model training is performed using a labeled dataset with severity annotations. This stage includes model compilation, where the loss function and optimization strategy are defined, followed by training through forward and backward passes with weight updates. In the final stage, the trained model is evaluated on previously unseen IoT vulnerability descriptions, producing categorical severity predictions (Low, Medium, High, or Critical). Model performance is assessed using standard evaluation metrics, including Accuracy, Precision, Recall, and F1-score, to measure effectiveness and reliability.

The core of the framework is a transformer-based language model that learns contextual representations of vulnerability descriptions. These representations capture semantic relationships across the entire input sequence and form the basis for downstream classification. Finally, a classification layer maps the learned representations to predefined severity categories, producing the final severity prediction.

3.3. DistilBERT-Based Representation Learning

To achieve an effective balance between predictive performance and computational efficiency, the proposed framework adopts fine-tuned DistilBERT as the backbone language model. DistilBERT is a distilled variant of the Bidirectional Encoder Representations from Transformers (BERT) architecture. DistilBERT was selected for this study because it is 40% smaller and 60% faster than the base BERT model while retaining approximately 97% of its predictive performance [13]. It preserves much of BERT's contextual representation capability while substantially reducing model size and inference cost through knowledge distillation. Shan et al. [20] demonstrate that DistilBERT reduces execution time by approximately 50% compared with BERT, while achieving slightly improved accuracy. These characteristics make the model well suited for large-scale analysis tasks and deployment in resource-constrained environments. Consequently, the framework satisfies the computational constraints inherent in real-time IoT vulnerability assessment.

Prior research into transformer-based models for vulnerability analysis has established a foundation for the field, yet several limitations persist [9–11,16,19]. Existing studies typically evaluate performance using IT-centric datasets, which may not accurately reflect the unique requirements of other sectors. Furthermore, these approaches prioritize computational efficiency as the primary metric for success. This focus often comes at the

expense of domain adaptation, leaving a gap in how these models transition to specialised environments like IoT security.

In contrast, this study adapts DistilBERT specifically to IoT-focused vulnerability severity prediction under data-scarce and imbalanced conditions. DistilBERT employs a transformer architecture with fewer layers and parameters than full BERT models, enabling faster training and inference without substantial degradation in accuracy. In this work, DistilBERT is used as a feature extractor that transforms each vulnerability description into a dense contextual embedding. These embeddings capture bidirectional semantic relationships between words and phrases, allowing the model to effectively interpret technical terminology, exploit descriptions, and impact statements commonly found in vulnerability reports. A specific Fine-tuning Layer is added on top. This layer is trained specifically to map the text descriptions to three severity levels: Critical, High, Medium, and Low.

The choice of DistilBERT is motivated by its balance between representational capacity and computational efficiency. Larger transformer models introduce higher computational overhead and an increased risk of overfitting when training data are limited, which reduces their practicality in operational IoT security settings that often lack GPU acceleration. By contrast, DistilBERT provides sufficient linguistic expressiveness for vulnerability description analysis while remaining compatible with CPU-only and resource-constrained deployment scenarios. Importantly, this study focuses on evaluating if an existing, efficient AI model can be successfully repurposed to identify security risks in smart devices. The study does not attempt to design a new type of AI architecture. Instead, it tests how well a streamlined transformer model handles the unique data of IoT vulnerabilities while operating under actual hardware and power limitations.

3.4. Severity Classification and Transfer Learning

The contextual embeddings generated by DistilBERT are passed to a fully connected classification layer that outputs the predicted severity class. This layer produces a probability distribution over the four predefined severity categories, Low, Medium, High, and Critical, using a Softmax function. The final severity label is determined by the class with the highest predicted probability.

During training, the entire model, including both the DistilBERT backbone and the classification head, is fine-tuned using labelled IoT vulnerability data. This transfer learning strategy enables the model to adapt general linguistic knowledge acquired during large-scale pretraining to the domain-specific characteristics of IoT vulnerability descriptions. As a result, the model learns to recognise severity-relevant indicators such as device-specific terminology, references to embedded systems, and context-dependent risk factors. Model optimisation is performed using a multi-class cross-entropy loss function, which encourages accurate classification across all severity levels. This formulation supports balanced learning and enables consistent severity prediction without reliance on manual scoring or handcrafted rules.

3.5. Advantages of the Proposed Approach

The proposed IoT vulnerability severity prediction framework offers several advantages over traditional and existing automated methods. First, it reduces reliance on manual CVSS scoring for severity categorisation, thereby mitigating subjectivity and improving scalability. Second, by leveraging contextual language representations, the framework captures nuanced semantic information that is difficult to encode using rule-based or feature-engineered approaches. Third, the use of DistilBERT provides an effective trade-

off between predictive performance and computational efficiency, making the approach suitable for large-scale and resource-constrained environments.

Most importantly, the framework is explicitly designed for IoT-focused vulnerability analysis. Unlike prior approaches that are predominantly evaluated on IT-centric datasets, this work demonstrates the feasibility of adapting a lightweight transformer model to IoT vulnerability severity prediction using an IoT-specific dataset. The results provide empirical evidence that transfer learning can effectively bridge the gap between general-purpose language models and domain-specific IoT security tasks.

4. Performance Evaluation

This section evaluates the effectiveness, robustness, and generalisation capability of the proposed IoT vulnerability severity prediction framework. The performance of the proposed model is evaluated using standard multi-class classification metrics.

4.1. Experimental Setup

This section outlines the experimental setup adopted for model training and evaluation, with emphasis on reproducibility and fair comparison across methods.

4.1.1. Computing Environment and Implementation

All experiments were conducted on a Windows 10 Enterprise system equipped with an 11th Generation Intel Core i7-1185G7 CPU operating at 3.00 GHz, 16 GB of RAM, and a 64-bit architecture. Due to the absence of GPU support, all training and evaluation were performed using CPU-based computation. This setup reflects realistic resource constraints commonly encountered in IoT security environments.

The implementation was developed using Python 3.12. The Hugging Face Transformers library [26] was used to fine-tune the DistilBERT model for sequence classification, while PyTorch 2.6.0 supported model training and optimisation. Evaluation metrics and cross-validation utilities were implemented using scikit-learn, and natural language pre-processing was supported using the NLTK library. This toolchain ensures consistency and reproducibility across all experiments.

To ensure reproducibility, all experiments were conducted using fixed random seeds controlling dataset sampling, model initialisation, data loading, and optimisation procedures. The same seed configuration was applied consistently across the proposed model and all baseline methods. All experiments were conducted using a fixed random seed to ensure reproducibility across stochastic model initialisation, optimisation, and data shuffling. For neural network and transformer-based models, the random seed controlled weight initialisation, dropout behaviour, mini-batch shuffling, and optimiser operations. For traditional machine learning models, the seed controlled stochastic components such as data shuffling and solver initialisation where applicable. In addition, cross-validation folds were generated deterministically using the same seed to ensure identical training and testing splits across all models. These controls ensure that performance differences reported in the study are attributable to model design rather than random variation. The reproducibility settings and random seed usage are now clearly stated in the experimental setup section of the revised manuscript.

4.1.2. Input Representation and Sequence Length Handling

All input texts were standardised to a maximum sequence length of 512 tokens. Approximately 19% of the samples exceeded this limit and were truncated using a consistent head-only truncation strategy applied uniformly across all severity classes. This approach follows standard practice for transformer-based models and ensures uniform

input constraints, while acknowledging that some information loss may occur for longer vulnerability descriptions.

4.2. Baseline Models

The proposed IoTDistilBERT framework is compared against a set of traditional ML and baseline deep learning models, including Decision Tree (DT), Neural Network (NN), and Support Vector Machine (SVM) classifiers. Traditional ML models have been widely used as baseline approaches in prior studies [12,16,20,21]. These baselines provide representative reference points for evaluating performance under classical text representation and learning paradigms. All classical ML baselines were implemented using TF-IDF representations with uni-gram and bi-gram features derived from the same preprocessed text used across models. Hyperparameters were tuned via cross-validation, with macro-F1 selected as the optimisation criterion to ensure balanced evaluation across severity classes. A direct comparison with standard BERT is outside the scope of this study, as the focus is on deployment feasibility and computational efficiency under resource-constrained conditions rather than maximising model capacity.

We also include CVSS-BERT [10] as a baseline, which is a standard BERT-based approach. This inclusion provides an explicit comparison with full-scale BERT models. The motivation for selecting CVSS-BERT is that it represents a well-established BERT-based method for vulnerability severity prediction and serves as a strong reference point for evaluating the trade-off between predictive performance and computational efficiency.

4.3. Dataset Description

In this section, we describe the dataset used to evaluate the proposed IoT vulnerability severity prediction framework, including its source, structure, severity labeling procedure, preprocessing, and balancing strategy.

4.3.1. VARIOt Dataset Overview

The proposed model was evaluated using an IoT-focused vulnerability dataset obtained from the VARIOt repository [15]. VARIOt is a structured vulnerability dataset designed to support automated IoT security analysis and severity prediction. Each record corresponds to a single IoT-related vulnerability and aggregates information from multiple sources into a unified format.

4.3.2. Vulnerability Domains

The dataset contained 26,788 vulnerabilities and spans vulnerability reports published between 2000 and 2024. It reflects the heterogeneity of real-world IoT ecosystems. Reported vulnerabilities cover device-level issues, firmware and operating system flaws, communication protocol weaknesses, and platform or application-layer components. This diversity introduces variation in language style and technical focus, providing a realistic evaluation setting for text-based severity prediction.

Figure 3 illustrates the distribution of vulnerabilities across different IoT-related categories. Device and hardware vulnerabilities constitute the largest proportion, accounting for approximately 40% of the total, highlighting the significant exposure at the physical device level. Platform and software-related vulnerabilities form the second largest group at around 25%, indicating persistent weaknesses in operating systems and application software used within IoT ecosystems. Firmware vulnerabilities represent roughly 18%, reflecting risks associated with embedded code and update mechanisms. In contrast, communication protocol vulnerabilities account for a much smaller share, at about 6%, suggesting relatively fewer reported issues at the protocol layer. Finally, unclassified vulnerabilities make up approximately 10% of the dataset, indicating cases where insufficient

information prevents precise categorisation. Overall, the distribution underscores that the majority of IoT vulnerabilities originate at the device, software, and firmware layers, reinforcing the need for focused security analysis and mitigation at these levels.

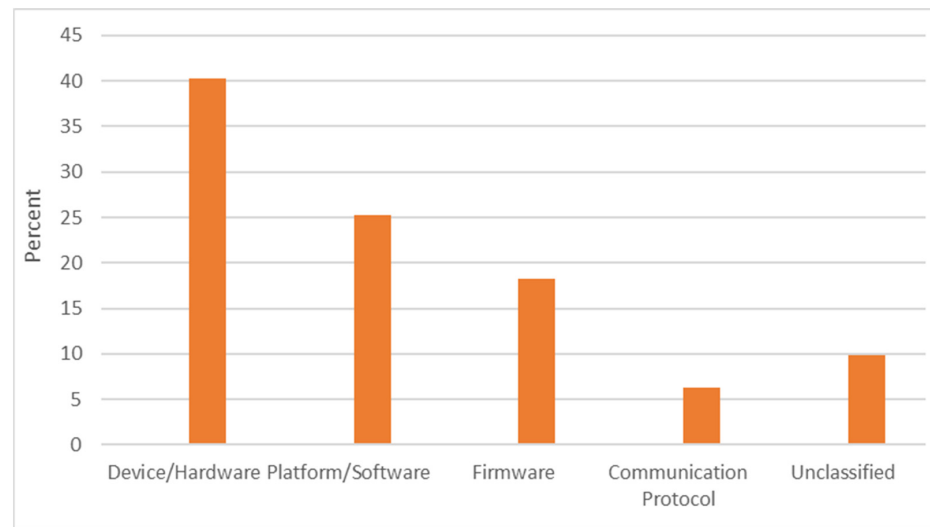


Figure 3. Distribution of vulnerabilities across different.

4.3.3. Dataset Schema and Fields

This subsection describes the schema and fields of the original dataset as defined in the VARIoT repository. An overview of the dataset schema is presented in Table 3, which illustrates the relationship between unstructured descriptions, structured fields, and auxiliary metadata as defined in the original dataset obtained from the VARIoT repository. Each record includes a unique vulnerability identifier, a textual vulnerability description, affected device information, IoT component categorisation, severity annotations, and supporting metadata such as reference URLs and disclosure details. Structured fields, including the CVE or VARIoT identifier and CVSS base score when available, support dataset organisation, filtering, and ground-truth labelling but are not required for the core text-based classification task.

These structured fields facilitate ground-truth labeling, filtering, and dataset management but are not required for the core text-based classification task. An overview of the dataset schema is provided in Table 3, illustrating the relationship between unstructured descriptions and structured metadata.

Table 3. Structural components of the VARIoT dataset.

Data Type	Value	Hyperparameter
Unstructured	Vulnerability Description, Impact Notes	Primary input for DistilBERT model.
Structured	CVE ID, CVSS Score	Used for ground-truth labeling and filtering.
Metadata	Source URL, Discovery Date	Auxiliary info for recorded duplication.

4.3.4. Preprocessing and Filtering

A custom Python script was developed to retrieve a set of vulnerability records via the VARIoT API and perform dataset preparation. For each vulnerability record, the corresponding textual description and its associated categorical severity label were extracted. The resulting dataset is defined as

$$D = \{(x_i, y_i)\}_{i=1}^N \quad (4)$$

where x_i represents the raw vulnerability description and $y_i \in \mathcal{Y}$ denotes the CVSS-derived severity level (see Equation (2)).

The model uses only the raw vulnerability description while the corresponding severity labels are used exclusively during training for loss computation and weight updates.

Text preprocessing was deliberately kept minimal in line with best practices for transformer-based models. Specifically, preprocessing was limited to basic text cleaning and formatting, such as removal of non-informative special characters and whitespace normalisation. No stopword removal, stemming, or lemmatization was applied, as such operations may interfere with subword tokenization and remove semantically useful information.

4.3.5. Dataset Balancing and Length Characteristics

To address severe class imbalance in the original dataset, a balanced subset of 4020 vulnerability records was constructed via random downsampling. Specifically, the number of samples in each severity class was reduced to match the size of the smallest class by randomly selecting records without replacement. The downsampling procedure was performed once using a fixed random seed to create a controlled experimental dataset suitable for fair and consistent comparative evaluation under CPU-only computational constraints. Algorithm 1 summarises the random downsampling procedure used to construct the balanced dataset.

Algorithm 1: Random Downsampling for Class Balancing

Input:

$D = \{(x_i, y_i)\}_{i=1}^N$ // Dataset of vulnerability description x_i and severity level $y_i \in \mathcal{Y}$
 $\mathcal{Y} = \{Low, Medium, High, Critical\}$ // set of severity classes
 seed = 42

Output:

D_{balanced} : class-balanced dataset

Begin:

1. Initialize the random number generator using seed.
2. Class-wise partitioning: For each severity class $c \in \mathcal{Y}$, define

$$D_c = \{(x_i, y_i) \in D \mid y_i = c\}, \forall c \in \mathcal{Y}$$

where D_c is the subset of samples in D belonging to class c .

3. Minority class size:

$$n = \min_{c \in \mathcal{Y}} |D_c|$$

where $|D_c|$ denotes the number of samples in class c .

4. Random downsampling: For each $c \in \mathcal{Y}$, construct

$$D_c^{\text{balanced}} = \text{RandomSample}(D_c, n)$$

by randomly selecting n samples from D_c without replacement.

5. Balanced dataset construction: Define

$$D_{\text{balanced}} = \bigcup_{c \in \mathcal{Y}} D_c^{\text{balanced}}$$

End

Random downsampling was chosen over oversampling to avoid introducing synthetic or duplicated vulnerability descriptions, which may distort the semantic structure of natural language text and artificially inflate model performance. In addition, oversampling techniques such as text-based data augmentation or SMOTE-style methods typically increase dataset size and training cost, which is inconsistent with the CPU-only and resource-constrained evaluation setting considered in this study. Downsampling provides a controlled and unbiased comparison across models by preserving original text samples while maintaining balanced class distributions. Although downsampling was adopted in this study for the reasons outlined above, future work will investigate oversampling and data augmentation strategies tailored for vulnerability text, including imbalance-aware learning methods, to better preserve real-world severity distributions while maintaining computational efficiency.

Figure 4 illustrates the length distribution of vulnerability descriptions in the dataset. IoT vulnerability reports are relatively long and information-dense, with an average length of approximately 210 words and a maximum exceeding 1100 words. This characteristic highlights the need for models capable of capturing long-range contextual dependencies in unstructured technical text and motivates the use of transformer-based architectures.

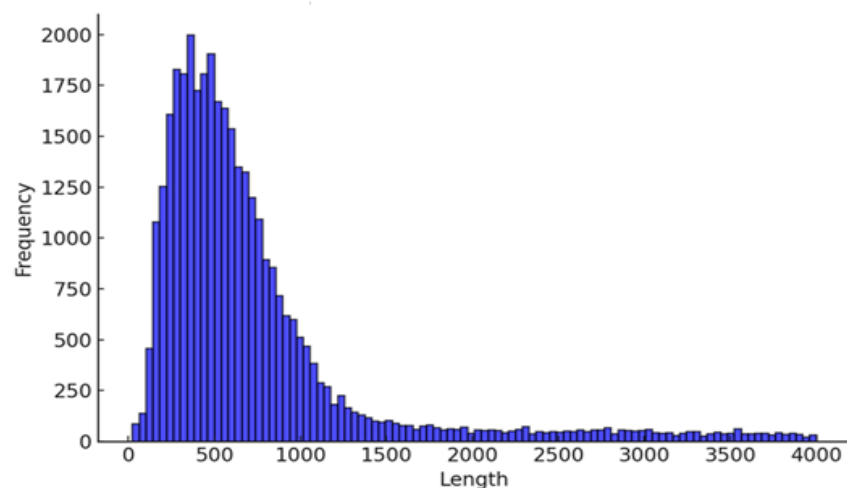


Figure 4. Length Distribution of Vulnerability Descriptions.

4.4. Training Strategy and Hyperparameter Optimization

To ensure robust performance estimation and reduce overfitting, stratified five-fold cross-validation was employed, consistent with established evaluation practices in statistical learning [27]. Stratification preserved the original class distribution across folds, providing an unbiased evaluation for each severity category. In each iteration, the model was trained on four folds and evaluated on the remaining fold, with results averaged across all folds. The hyperparameters used to train the model are summarised in Table 4.

Table 4. Key hyperparameters used in the training process.

Hyperparameter	Value	Hyperparameter	Value
Pre-trained model	DistilBERT	Number of training epochs	10
Training batch size	8	Gradient accumulation steps	2
Learning rate	3×10^{-5}	Evaluation batch size	32
Optimiser	AdamW	Label smoothing factor	0.1
Weight decay	0.0001	Cross-validation strategy	Stratified 5-fold
Warm-up steps	250	Tokenization	WordPiece
Hardware	CPU-only	Maximum sequence length	512
		Loss function	Cross-entropy loss

Hyperparameter optimisation was performed using the Optuna framework. Multiple trials were conducted to identify an optimal configuration that balances convergence stability and computational efficiency. The final model configuration consisted of 10 training epochs, a batch size of 8 for training and 32 for evaluation, a learning rate of 3×10^{-5} , and a weight decay of 0.0001. Gradient accumulation and label smoothing were applied to stabilise training under limited batch sizes. The model achieving the highest validation accuracy was automatically selected at the end of training. Hyperparameter optimisation was conducted within the training data of each cross-validation fold using Optuna. The evaluation protocol avoids test-set leakage but does not implement full nested cross-validation, which is left for future work due to computational constraints.

4.5. Evaluation Metrics

Model performance was evaluated using four widely adopted multi-class classification metrics [9,20]: Accuracy, Precision, Recall, and F1-score. These metrics are commonly used to assess classification performance in security-related prediction tasks. In this paper, they are computed using the macro-averaging strategy, where metrics are first calculated independently for each severity class and then averaged equally across classes. This choice ensures that all severity levels contribute equally to the final score, which is appropriate given the use of a class-balanced dataset.

Accuracy measures the proportion of correctly classified instances across all severity classes. Accuracy is calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

where TP and TN represent true positives and true negatives, respectively. FP and FN denote false positives and false negatives, respectively.

Precision evaluates the reliability of predicted severity labels by measuring the proportion of correctly identified positive instances among all predicted positives. It is computed as

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Recall (Sensitivity) measures the model's ability to correctly identify true positive instances, defined as cases where a vulnerability is correctly assigned to its ground-truth severity class. Recall is calculated as

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

The F1-score, defined as the harmonic mean of Precision and Recall, provides a balanced measure of classification performance, particularly in the presence of class imbalance:

$$F1 - score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \quad (8)$$

It is particularly useful when you have an uneven class distribution, as it provides a more balanced view of performance than simple accuracy alone.

The relative importance of Precision and Recall depends on the deployment context. In safety-critical IoT systems, high Recall for the Critical severity class is essential to minimise the risk of overlooking high-impact vulnerabilities. Conversely, environments with limited response capacity may prioritise Precision to reduce false alarms and analyst workload. The F1-score, defined as the harmonic mean of Precision and Recall, provides a balanced measure of performance and is particularly well suited to imbalanced security datasets.

4.5.1. Cross-Validation Protocol

All evaluation metrics, including Accuracy, Precision, Recall, and F1-score, were computed independently for each fold during cross-validation. The dataset was partitioned into five mutually exclusive folds, with four folds used for training and one fold used for testing in each iteration. This process was repeated until each fold had been used once as the test set.

Final performance results are reported as the mean value $\bar{M}$ across the five folds, providing a reliable estimate of generalisation performance on unseen vulnerability descriptions. Let k denote the number of folds, where $k = 5$, and let M_j represent the value of a given evaluation metric obtained on fold j . The mean performance metric $\bar{M}$ is computed as

$$\bar{M} = \frac{1}{k} \sum_{j=1}^k M_j \quad (9)$$

This cross-validation-based averaging strategy mitigates the effects of favourable or unfavourable data splits and ensures that reported results reflect consistent model behaviour rather than fold-specific outcomes. Formal statistical significance testing was not conducted and is left for future work as larger IoT-specific datasets become available.

4.5.2. Error Reduction Rate

The Error Reduction Rate (ERR) is used to quantify the relative improvement of the proposed model over a baseline by measuring the reduction in classification error. ERR is particularly useful for comparing models with different baseline performance levels, as it normalises improvement with respect to the baseline error. The metric is defined as

$$ERR = \frac{BL_{Error} - PM_{Error}}{BL_{Error}} \times 100 \quad (10)$$

where BL_{Error} represents baseline error and PM_{Error} represents proposed model error.

In this study, classification error is derived from model accuracy. Specifically, the BL_{Error} is computed as

$$BL_{Error} = 1 - Accuracy_{baseline} \quad (11)$$

where the baseline model may be CVSS-BERT or a traditional classifier such as SVM. Similarly, the proposed model Error is computed as

$$PM_{Error} = 1 - Accuracy_{IoTDistilBERT} \quad (12)$$

A higher ERR value indicates a greater reduction in error achieved by the proposed model relative to the baseline, providing an intuitive measure of performance improvement in terms of misclassification reduction.

4.5.3. Quantifying Performance Improvement

To measure how much IoTDistilBERT outperforms the baseline models, we use a standard Percentage Increase formula. This calculation determines the relative gain in the F-Score between the proposed model and its competitors. The formula is as follows:

$$Percentage\ increase = \frac{IoTDistilBERT - Baseline}{Baseline} \times 100 \quad (13)$$

This metric provides an intuitive measure of relative performance improvement and facilitates direct comparison across different baseline approaches.

Precision, recall, and F1-score are computed in a multi-class setting using a weighted averaging scheme, where class-specific metrics are weighted by the number of samples (support) in each class within the validation fold. This weighted formulation ensures that classes with larger representation contribute proportionally to the overall performance, while still reflecting residual class imbalance. In addition to weighted metrics, the macro-averaged F1-score is also calculated and reported. The macro F1-score is obtained by computing the F1-score independently for each class and then averaging these scores equally across all classes, irrespective of class frequency.

All metrics are computed independently for each fold of the stratified five-fold cross-validation procedure using predictions on the held-out validation set. For each fold, class predictions are obtained via the argmax of model logits, and precision, recall, and F1-scores are derived from the resulting confusion matrix.

Final reported performance is obtained by aggregating metrics across folds using the arithmetic mean, and fold-level variability is summarised using the standard deviation and observed minimum–maximum range. This aggregation provides both a central performance estimate and an indication of stability across different train–validation splits.

5. Results

This section presents the experimental results obtained for the proposed IoT vulnerability severity prediction framework and provides a comparative analysis against baseline ML and deep learning models. The evaluation focuses on standard multi-class classification metrics, including accuracy, precision, recall, and F1-score, computed using stratified cross-validation to ensure robust performance estimation. In addition to aggregate metrics, confusion matrix analysis is reported to highlight the model's behaviour across different severity levels. Comparative results are used to assess the effectiveness of the proposed approach relative to existing methods, with particular emphasis on predictive reliability, generalisation under data-scarce conditions, and suitability for deployment in resource-constrained IoT security environments. The model produces class probability distributions via a softmax output layer. For all reported results, final predictions are obtained using a standard maximum-probability (argmax) decision rule to ensure consistent evaluation across models. Also, performance figures are interpreted in terms of relative trends rather than absolute superiority.

5.1. Accuracy

This subsection reports the accuracy performance of the proposed IoT vulnerability severity prediction model and compares it against baseline ML and deep learning approaches. Accuracy is used as an initial indicator of the model's overall ability to correctly classify vulnerability descriptions across the four severity categories. As shown in Figure 5, the proposed IoTDistilBERT framework achieved a mean classification accuracy of approximately 91.5% across the five cross-validation folds. This result indicates that the majority of IoT vulnerability descriptions were correctly mapped to their corresponding severity levels. The accuracy values remained consistently high across folds, with only minor variations, suggesting stable learning behaviour and limited sensitivity to specific data partitions.

Comparative evaluation shows that the proposed model outperformed all baseline approaches in terms of accuracy. Traditional ML classifiers, such as Support Vector Machines, Naïve Bayes, and Decision Trees, exhibited noticeably lower accuracy values, reflecting their limited ability to capture complex semantic relationships in unstructured vulnerability descriptions. Similarly, baseline neural network models achieved moderate improvements over traditional methods but remained inferior to transformer-based approaches. Among deep learning baselines, transformer-based models demonstrated superior accuracy

compared to non-transformer architectures. In particular, the proposed model achieved accuracy comparable to or exceeding that of larger transformer-based frameworks, while requiring fewer parameters and reduced computational resources. This result highlights the effectiveness of leveraging contextual language representations through a lightweight transformer architecture. The accuracy results demonstrate that the proposed IoTDistilBERT framework provides a reliable and efficient solution for severity classification, achieving strong performance while maintaining computational efficiency.

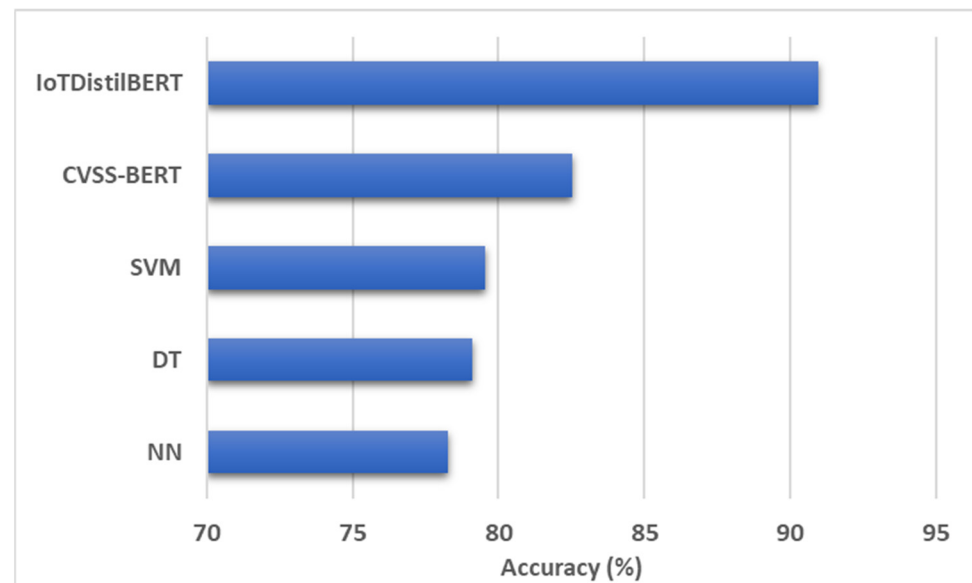


Figure 5. Comparison of mean accuracy across models using stratified 5-fold cross-validation.

It is also noteworthy that the reported accuracy was achieved under CPU-only training and evaluation conditions, without GPU acceleration. This reinforces the practicality of the proposed approach for deployment in real-world IoT security environments, where access to high-performance computing resources may be limited. The consistently high accuracy across folds further suggests that the model generalises well to unseen IoT vulnerability descriptions.

5.2. Precision

This subsection presents the precision performance of the proposed IoT vulnerability severity prediction model across the four severity classes. Precision is a critical evaluation metric in vulnerability assessment, as it measures the proportion of correctly predicted severity labels among all predictions made for a given class. High precision is particularly important for high-severity categories, as it reduces false positives that could lead to inefficient allocation of security resources. Figure 6 shows the precision results of the models.

The proposed IoTDistilBERT framework achieved consistently high precision scores across all five cross-validation folds, with a mean precision exceeding 91%. This indicates that the majority of severity predictions produced by the model were accurate and reliable. Precision values showed limited variation across folds, demonstrating stable predictive behaviour and robustness to changes in training and validation splits. When compared with baseline methods, the proposed model exhibited superior precision performance. Traditional ML classifiers showed lower precision, particularly for higher severity classes, reflecting their difficulty in distinguishing subtle semantic differences in vulnerability descriptions. Baseline deep learning models improved upon traditional approaches but still produced a higher number of false-positive predictions compared to the proposed transformer-based framework.

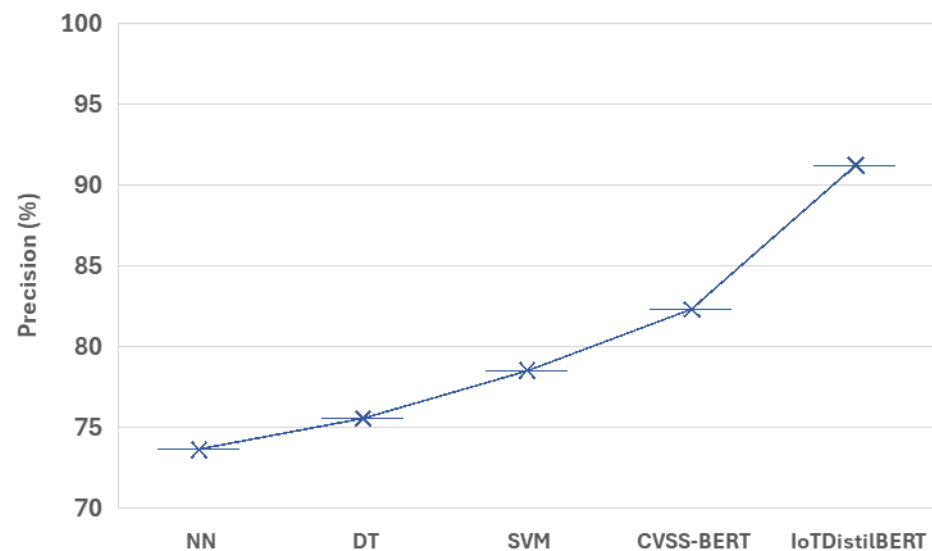


Figure 6. Comparison of mean precision across models using stratified 5-fold cross-validation.

Notably, the proposed model achieved strong precision for the High and Critical severity categories. This is a key result in the context of IoT security, where overestimating severity can lead to unnecessary remediation efforts and alert fatigue. The high precision achieved by the model suggests that it can reliably identify genuinely severe vulnerabilities without generating excessive false alarms. The precision results confirm that the proposed IoTDistilBERT approach provides accurate and trustworthy severity predictions. The ability to maintain high precision while operating under data-scarce conditions and CPU-only computation further highlights the suitability of the proposed framework for practical IoT vulnerability management.

5.3. Recall

This subsection reports the recall performance of the proposed IoT vulnerability severity prediction model across the four severity categories. Recall measures the proportion of actual instances of a given severity class that are correctly identified by the model. In the context of vulnerability management, high recall is particularly important for High and Critical severity vulnerabilities, as failing to identify these cases may leave significant security risks unaddressed. Figure 7 shows the recall results of the models.

The proposed IoTDistilBERT framework achieved a mean recall of approximately 92% across the five cross-validation folds. Recall values were consistently high across folds, with limited variability, indicating that the model effectively identifies relevant vulnerabilities across different data partitions. This stability suggests that the model generalises well to unseen IoT vulnerability descriptions. Compared to baseline methods, the proposed model demonstrated superior recall performance. Traditional ML classifiers exhibited noticeably lower recall, particularly for higher severity classes, reflecting a tendency to miss complex or context-dependent vulnerability descriptions. Baseline deep learning models improved recall relative to traditional approaches but still underperformed compared to the proposed transformer-based framework.

The recall results for the High and Critical severity categories are particularly notable. The model successfully identified the majority of vulnerabilities in these categories, minimising false negatives and reducing the risk of overlooking severe security threats. This characteristic is essential for IoT security applications, where undetected high-impact vulnerabilities can have serious operational and safety consequences. The recall results indicate that the proposed approach provides comprehensive coverage of vulnerability

severity classes, ensuring that critical threats are identified reliably. The consistently high recall achieved under data-scarce conditions and CPU-only execution further demonstrates the robustness and practicality of the proposed framework.

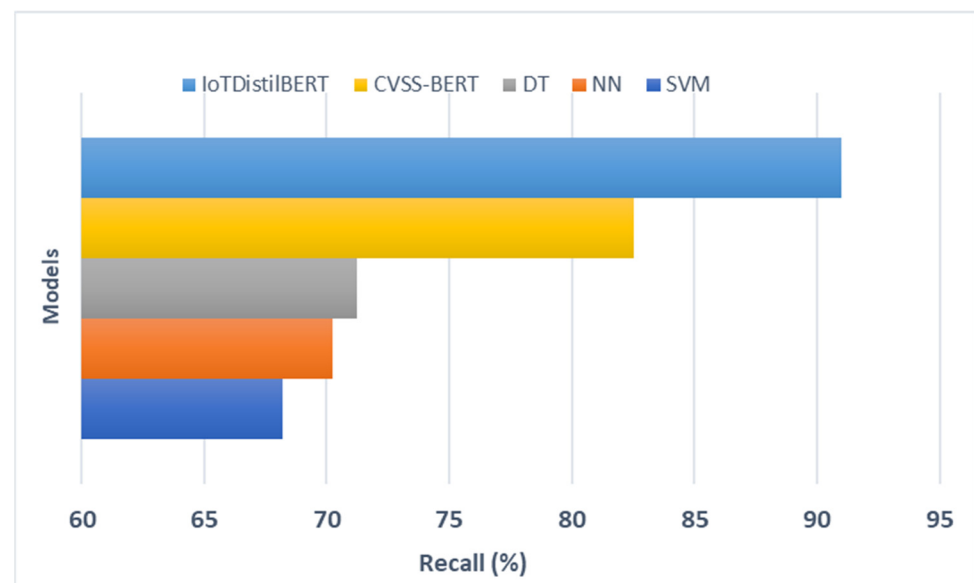


Figure 7. Comparison of mean recall across models using stratified 5-fold cross-validation.

5.4. Prediction of F1-Score

This subsection presents the F1-score performance of the proposed IoT vulnerability severity prediction model. Figure 6 shows the F1-score results of the models. The F1-score, defined as the harmonic mean of Precision and Recall, provides a balanced evaluation of classification performance by accounting for both false positives and false negatives. As such, it is particularly suitable for vulnerability severity prediction tasks involving class imbalance and asymmetric risk considerations.

Figure 8 clearly shows that the two transformer-based models (IoTDistilBERT and CVSS-BERT) occupy nearly half of the total performance “space” in the comparison. The proposed IoTDistilBERT framework achieved a mean F1-score of approximately 91.8% across the five cross-validation folds. The F1-score values remained consistently high across all folds, with only minor variation, indicating stable and balanced classification performance across different training and validation splits.

In comparison to baseline methods, the proposed model demonstrated a clear advantage in terms of F1-score. Traditional ML classifiers yielded substantially lower F1-scores, reflecting their limited ability to balance precision and recall in complex textual classification tasks. Baseline deep learning models showed improved F1-scores relative to traditional methods but remained consistently below the performance of the proposed transformer-based approach.

The strong F1-score performance indicates that the proposed framework effectively balances the trade-off between identifying severe vulnerabilities and avoiding excessive false alarms. This balance is especially important in IoT security contexts, where both missed high-risk vulnerabilities and unnecessary remediation actions can have significant operational consequences. The F1-score results confirm that the proposed IoTDistilBERT model provides robust and reliable severity predictions across all classes. The consistency of F1-scores across folds further supports the model’s ability to generalise under data-scarce conditions and CPU-only execution.

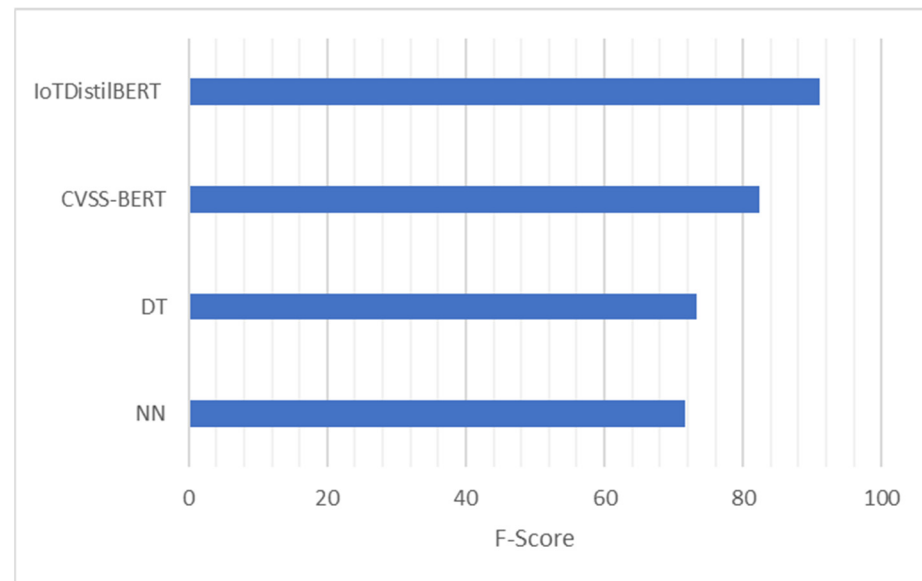


Figure 8. Comparison of mean F1-Score across models using stratified 5-fold cross-validation.

5.5. Performance Improvement Analysis

The empirical results demonstrate that IoTDistilBERT consistently outperforms both traditional ML baselines and transformer-based models across all evaluated metrics. Performance gains were analysed by computing the relative improvement in F1-score, Accuracy, Precision, and Recall. Table 5. Summarises of the relative performance gains of IoT-DistilBERT over traditional and transformer-based baselines. The results indicate that IoTDistilBERT achieves double-digit percentage improvements over all baseline models for each metric.

Table 5. Summary of the relative performance gains of IoTDistilBERT over traditional and transformer-based baselines.

Metric	Baseline Score (%)			
	CB	DT	NN	SVM
F-Score	10.98	24.66	28.17	28.17
Accuracy	10.30	15.04	16.37	14.47
Precision	10.81	20.63	23.91	16.18
Recall	10.42	27.77	29.59	33.58

In comparison with the transformer-based CVSS-BERT model, IoTDistilBERT maintains an improvement of approximately 10% to 11% across all metrics. The performance gap increases substantially when compared with traditional classifiers such as Decision Tree (DT), Neural Network (NN), and Support Vector Machine (SVM). These results highlight the effectiveness of lightweight transformer-based representations for vulnerability severity classification.

The most substantial gains are observed in Recall, reflecting improved detection capability and a stronger ability to correctly identify positive instances. The largest improvement is achieved against the SVM baseline, where Recall increases by 33.58%. Similarly, the 28.17% improvement in F1-score over both SVM and NN baselines underscores the advantage of leveraging pre-trained transformer representations for complex semantic feature extraction. Since the F1-score captures the harmonic mean of Precision and Recall, these gains indicate that the proposed fine-tuning strategy and DistilBERT's architectural optimisation are particularly effective for this dataset and task.

5.6. Evaluation of Error Reduction Rate

The performance of the proposed IoTDistilBERT model is evaluated through a comparative analysis of error reduction across multiple metrics. This section synthesises the performance gains of IoTDistilBERT by analysing the transition from baseline error rates to the proposed model's enhanced reliability. By shifting the focus from raw scores to error reduction, we can better quantify the technical impact of the architectural optimizations.

5.6.1. Relative Error Reduction for Accuracy

The first phase of evaluation compares IoTDistilBERT against several baseline architectures, including CVSS-BERT, SVM, DT, and NN. As illustrated in Figure 9, the proposed model consistently achieves significant error reduction across all baselines. The error reduction rate is calculated by comparing the number of misclassifications. It shows the percentage of “missed” or “wrong” predictions that IoTDistilBERT successfully corrects. A primary observation from this data is the substantial level of error elimination achieved. In all tested cases, the error reduction rate significantly exceeds the baseline error rate, with improvements ranging from 48.57% to 58.72%. This demonstrates a high level of performance consistency across different comparative scenarios.

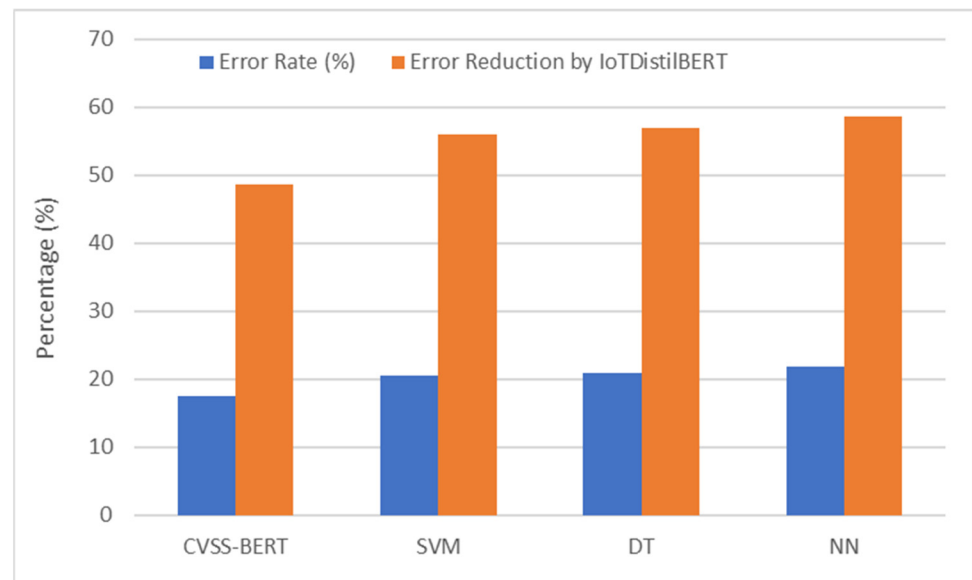


Figure 9. Comparison of Baseline Error Rates and Relative Error Reduction achieved by IoTDistilBERT.

When compared to the CVSS-BERT transformer model, IoTDistilBERT reduces the total number of errors by nearly half. In a practical security environment, this means that for every 100 threats that CVSS-BERT might fail to identify correctly, our model would likely catch approximately 48 of them. The improvement is even more pronounced when looking at traditional ML models like SVM and Neural Networks. In these cases, IoTDistilBERT achieves an error reduction of over 56% misclassification. This demonstrates that the optimized transformer architecture is significantly more robust at handling the complexities of IoT data than standard algorithmic approaches.

Furthermore, a significant transformer to transformer gain is evident. Even when measured against the next best performer, CVSS-BERT, the IoTDistilBERT model reduces the remaining errors by nearly half at 48.57%. The most dramatic gains are observed when compared to conventional ML approaches. Specifically, against NN and DT, the proposed model corrects over 56% of the misclassifications made by those baselines. Ultimately, this result demonstrates that IoTDistilBERT is not merely an incremental improvement. Instead,

it serves as a robust solution that fundamentally increases detection reliability across a diverse range of model comparisons.

5.6.2. Absolute Error Rates for Precision, Recall and F1-Score

To further validate the model's effectiveness in a cybersecurity context, the evaluation extended to Precision, Recall, and the F1-Score. These metrics provide insight into the model's ability to balance the reduction in false alarms with the detection of actual threats. Figure 10 compares the absolute error rates of IoTDistilBERT and CVSS-BERT for Precision, Recall, and F1-score. The figure illustrates the reduction in classification errors achieved by the proposed IoTDistilBERT model relative to the CVSS-BERT baseline. The error rate is defined as the complement of each metric (i.e., $1 - \text{metric}$) and represents the proportion of incorrect predictions.

As shown in Figure 10, IoTDistilBERT consistently achieves lower error rates across all three metrics. For Precision, the error rate decreases from 17.7% with CVSS-BERT to 8.8%, indicating a substantial reduction in false alarms. Similarly, the Recall error rate drops from 17.5% to 8.9%, demonstrating a marked improvement in the detection of previously missed high-risk vulnerabilities. Corresponding reductions are also observed for the F1-score, reflecting a more balanced overall classification performance. These results confirm that IoTDistilBERT not only improves predictive accuracy but also significantly reduces misclassification rates, which is critical in IoT security contexts where both false positives and false negatives can incur high operational costs.

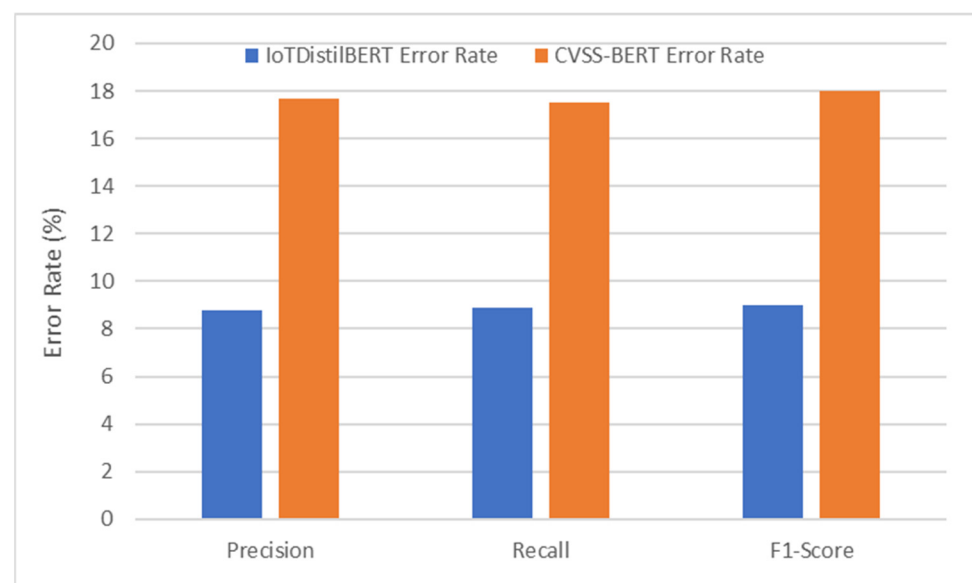


Figure 10. Comparison of absolute error rates for Precision, Recall and F1-Score between CVSS-BERT and IoTDistilBERT.

5.6.3. Magnitude of Relative Error Reduction

This section quantifies the proportion of baseline errors eliminated by the proposed model. Figure 11 illustrates the ERR achieved by IoTDistilBERT relative to the CVSS-BERT baseline for Precision, Recall and F1-Score. The dashed reference line indicates a 50% error reduction threshold, providing a clear benchmark for interpreting the magnitude of improvement.

As shown in Figure 11, IoTDistilBERT achieves an ERR of 50.28% for Precision, exceeding the 50% threshold, and 49.14% for Recall, which closely approaches this benchmark. These values indicate that approximately half of the classification errors produced by the CVSS-BERT model are eliminated by the proposed approach.

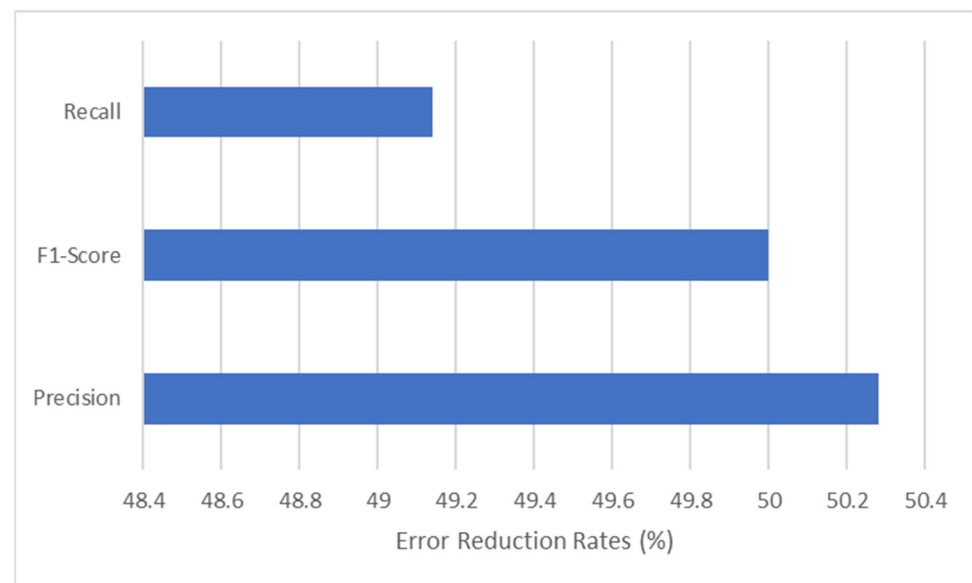


Figure 11. Magnitude of Relative Error Reduction achieved by IoTDistilBERT over CVSS-BERT.

The similar ERR values observed for Precision and Recall demonstrate that the performance gains are well balanced. Specifically, the proposed architecture reduces false alarms while simultaneously improving the detection of previously missed vulnerabilities. This balanced error reduction is particularly important in IoT security contexts, where both false positives and false negatives can lead to significant operational and security consequences. Overall, the ERR results confirm that IoTDistilBERT substantially improves reliability over the strongest transformer-based baseline and achieves consistent error reduction across key evaluation dimensions.

5.7. Confusion Matrix Analysis

Figure 12 presents the confusion matrix for IoT vulnerability severity prediction using the proposed IoTDistilBERT model. While this study emphasises overall severity classification reliability, Figure 12 provides a class-level view of prediction behaviour, including for High and Critical severity categories. The confusion matrix provides a detailed view of classification behaviour across the four severity categories: Low, Medium, High, and Critical. The model demonstrates strong performance in the Low severity category, correctly classifying 930 out of 1005 instances, corresponding to a class-specific recall of approximately 92.5%. The most common misclassification in this category involved confusion with the adjacent Medium severity level, where 60 instances were incorrectly labelled. Only a small number of Low severity vulnerabilities were misclassified as High (10 instances) or Critical (5 instances), indicating a low rate of severe overestimation. Similarly, the Medium severity category exhibits robust performance, with 895 correctly classified instances out of 1005. The majority of errors in this category involved misclassification as High severity (45 instances), while fewer cases were misclassified as Low (55 instances) or Critical (10 instances). This pattern suggests that misclassifications tend to occur between adjacent severity levels rather than across distant categories.

For the High severity category, the model correctly identified 910 out of 1005 instances. Most misclassifications involved confusion with the neighbouring Medium (42 instances) and Critical (45 instances) categories. Only a very small number of High severity vulnerabilities were misclassified as Low (8 instances), indicating a low incidence of extreme underestimation. The Critical severity category achieved the highest classification performance, with 942 correctly classified instances out of 1005. The majority of errors in this

category involved misclassification as High severity (50 instances), while only a negligible number of Critical vulnerabilities were misclassified as Medium (10 instances) or Low (3 instances). This result is particularly important from a security perspective, as it minimises the risk of failing to identify high-impact vulnerabilities.

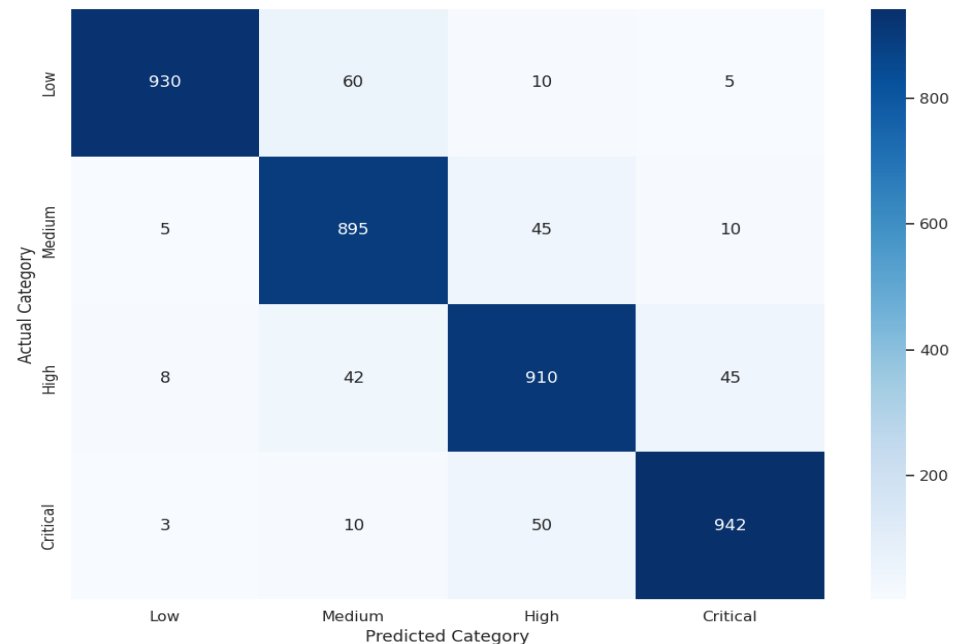


Figure 12. Confusion matrix for IoT vulnerability severity prediction.

The confusion matrix reveals a consistent neighbouring-class error pattern, where misclassifications predominantly occur between adjacent severity levels. Extreme misclassifications, such as labelling Critical vulnerabilities as Low, are rare. This error distribution, combined with the previously reported cross-validation accuracy of approximately 91.5%, indicates that the proposed model is both accurate and conservative in its predictions, making it well suited for practical IoT vulnerability severity assessment.

Inspection of the confusion matrix indicates that most misclassifications occur between adjacent severity categories, reflecting inherent ambiguity in vulnerability descriptions. From a triage perspective, such errors are less costly than confusions between non-adjacent classes, which are comparatively rare.

6. Discussion

The work presented in this paper positions itself at the intersection of automated vulnerability severity prediction, transfer learning, and IoT-focused security analysis. This study focuses on severity level prediction, which serves as a prerequisite for many downstream vulnerability management tasks, including prioritisation. This subsection discusses the predictive performance and generalisation behaviour of the proposed framework.

6.1. Predictive Performance and Generalisation

The experimental results demonstrate that the proposed IoT vulnerability severity prediction framework effectively captures semantic information embedded in unstructured vulnerability descriptions and translates it into accurate severity classifications. The consistently strong performance across evaluation metrics indicates that contextual language representations are well suited for modelling the nuanced technical narratives commonly found in IoT vulnerability reports.

Transfer learning plays a central role in this performance. By fine-tuning a pre-trained language model on IoT-specific vulnerability data, the proposed approach generalises effectively despite the limited size of the labelled dataset. Although minor performance variability is observed across cross-validation folds, the overall stability of the results suggests robust generalisation and limited sensitivity to data partitioning. This behaviour is particularly important in IoT security contexts, where available training data is often scarce and heterogeneous.

6.2. Comparison with Baseline Methods

Comparative evaluation highlights the advantages of the proposed DistilBERT-based model over conventional ML classifiers and baseline deep learning approaches. Across all reported metrics, the proposed framework achieves consistently stronger performance. Notably, improvements in Recall and F1-score indicate that the model is effective at identifying high-severity vulnerabilities while maintaining a balanced error profile. This characteristic is critical for vulnerability management, as it reduces the likelihood of overlooking high-risk threats without generating excessive false positives that could overwhelm security teams. The observed performance gains reflect the ability of contextual transformer-based representations to capture semantic relationships that are not readily accessible to feature-based or keyword-driven models.

The empirical results demonstrate that IoTDistilBERT consistently outperforms both traditional ML baselines and specialized transformer models across all evaluated metrics. The most notable improvement is observed in Recall, where IoTDistilBERT achieves a 33.58% increase over the SVM baseline. This high recall rate is critical in IoT security contexts because it minimizes the risk of undetected threats. Furthermore, the model maintains a significant lead in Precision, showing a 23.91% improvement over Neural Networks. This suggests that our approach not only identifies more relevant instances but also maintains a higher level of accuracy in its positive predictions. Even when compared to the CVSS-BERT model, IoTDistilBERT shows a steady gain of approximately 10% to 11% across F-Score, Accuracy, and Precision. These results indicate that the architectural optimizations in IoTDistilBERT are highly effective at capturing the nuances of IoT data. The performance gains are likely driven by the model's ability to capture contextual semantics of technical terms and early severity cues in vulnerability descriptions, combined with the robustness of transfer learning under data-scarce IoT conditions.

6.3. Analysis of Error Rate Reduction

The error reduction analysis provides additional insight into the practical reliability gains achieved by IoTDistilBERT beyond conventional performance metrics. While accuracy and F1-score quantify overall classification quality, the Error Reduction Rate (ERR) highlights how effectively the proposed model reduces misclassification relative to a strong transformer-based baseline. The observed reduction in absolute error rates for both Precision and Recall indicates that IoTDistilBERT substantially decreases false alarms while simultaneously improving the detection of previously missed vulnerabilities. Importantly, the magnitude of error reduction is highly consistent across Precision and Recall, suggesting that the proposed architectural and fine-tuning choices do not bias the model toward either conservative or aggressive detection behaviour.

The near-equal ERR values for Precision (50.28%), Recall (49.14%), and F1-score (50.0%) are particularly significant. This consistency indicates that performance improvements are systematic rather than metric-specific, reflecting a balanced enhancement in overall classification reliability. In practical IoT security contexts, such balance is important, as both excessive false positives and missed high-severity vulnerabilities can undermine the

trustworthiness of automated analysis systems. From a deployment perspective, reducing classification error by approximately half relative to a state-of-the-art BERT-based model highlights the robustness of the proposed approach under high-volume and heterogeneous IoT conditions. These improvements increase confidence in automated severity classification while reducing the need for manual inspection, particularly in resource-constrained or real-time analysis settings.

In summary, the ERR analysis reinforces the conclusion that IoTDistilBERT offers not only higher predictive performance but also meaningful gains in reliability and robustness, which are critical requirements for practical IoT vulnerability severity assessment.

6.4. Computational Efficiency and Lightweight Design

The performance achieved by the proposed framework is particularly significant given the computational constraints under which the experiments were conducted. All training and inference were performed using CPU-only resources on a relatively small IoT-focused dataset. These conditions reflect realistic deployment environments in many IoT security settings. The use of DistilBERT contributes directly to this efficiency. Compared with standard BERT architectures, DistilBERT employs fewer layers and parameters, resulting in reduced computational overhead and faster inference. Empirically, the model achieved an evaluation throughput of 5.17 samples per second with a total runtime of 78.17 s. These results demonstrate that the proposed framework can operate effectively without specialised GPU hardware, supporting near-real-time analysis under resource-constrained conditions.

6.5. Operational Implications for IoT Vulnerability Management

From an operational perspective, the combination of strong predictive performance and efficient inference highlights the feasibility of deploying the proposed approach in IoT vulnerability management contexts. Efficient inference enables timely severity assessment from textual vulnerability descriptions, which is particularly relevant in IoT ecosystems where devices are widely distributed, resource-constrained, and often difficult to update. In practical settings, vulnerability management involves asymmetric risk, as underestimating high-severity vulnerabilities may have greater consequences than conservative overestimation. Although a standard decision rule is used for evaluation in this study, the probabilistic outputs produced by the proposed model allow for post hoc calibration and class-specific thresholding if required in downstream applications. For example, conservative thresholds for the Critical class could be applied to reduce underestimation risk without retraining the model. These observations illustrate how the proposed framework can be integrated into broader IoT security workflows, while the primary contribution of this work remains the evaluation of lightweight Transformer-based severity classification under realistic deployment constraints.

6.6. Threats to Validity

This study is subject to several limitations that should be considered when interpreting the results. First, vulnerability severity labels are derived from discretised CVSS base scores using fixed thresholds. While this approach follows standard CVSS guidance, it may obscure finer-grained differences in risk and introduces sensitivity to boundary definitions. Second, to address severe class imbalance, dataset balancing was performed via downsampling. Although this enables stable training and fair evaluation across classes, it may reduce representativeness of real-world severity distributions and introduce dependence on a particular sampled subset. Third, all inputs were constrained to a maximum sequence length, and descriptions exceeding this limit were truncated. While truncation was applied consistently across classes, some information loss is unavoidable and may affect classification of longer vulnerability reports.

We note that similarity-aware cross-validation was not applied in this study, and near-duplicate textual descriptions, if present across folds, may pose a potential internal validity threat. This limitation is not unique to our work and is common in prior text-based vulnerability severity prediction studies [7–12,16–21]. These studies typically report random cross-validation or train/test splits without explicit similarity-aware controls.

Finally, hyperparameter optimisation was conducted within cross-validation folds but did not employ a fully nested cross-validation scheme. Although care was taken to avoid test-set leakage, this design choice may result in optimistic performance estimates. Future work will address these limitations through imbalance-aware training, length-sensitive modelling, and more rigorous evaluation protocols as larger datasets become available.

7. Conclusions and Future Directions

This study demonstrates the feasibility of lightweight transformer-based models for IoT vulnerability severity prediction under data and computational constraints. While the primary contribution lies in severity level prediction, the proposed approach can serve as a foundation for several downstream vulnerability management scenarios. For example, it may be used as a triage pre-filter to flag potentially high-severity IoT vulnerabilities for closer inspection, as an analyst support tool for reviewing large volumes of vulnerability disclosures or as a conservative severity screening mechanism in which stricter thresholds reduce the risk of underestimating Critical vulnerabilities. These scenarios are illustrative and are not design objectives of the proposed framework. Several limitations should be noted. Misclassification of high-impact vulnerabilities may have disproportionate consequences under data-scarce conditions. Model performance also depends on the quality and consistency of textual vulnerability descriptions, which can vary across IoT repositories. In addition, similarity-aware cross-validation was not applied, and near-duplicate textual descriptions across folds may constitute a potential internal validity threat.

Future work will focus on improving model transparency through explainable AI techniques and exploring domain-adaptive pretraining using IoT-specific technical corpora. Extensions to ordinal and fine-grained or continuous CVSS score prediction will also be investigated as larger and more consistently annotated IoT datasets become available. Deployment-oriented enhancements, including edge-based inference, integration with automated vulnerability management pipelines, and comprehensive profiling of inference latency and memory consumption, represent further directions. In addition, future work will explore imbalance-aware learning strategies that avoid dataset downsampling, such as cost-sensitive learning, class-weighted loss functions, and focal loss. More advanced similarity-aware splitting strategies are left for future work as larger IoT-specific datasets become available. Finally, systematic ablation studies, stronger linear baselines, and length-sensitive modelling approaches, including sliding-window segmentation and stratified performance analysis by input length, will be examined as data scale and computational resources permit. Statistical robustness analysis using paired tests or bootstrap-based confidence estimation is also left for future investigation.

Author Contributions: Conceptualisation, J.A. and S.A.B.; methodology, J.A.; software, S.A.B.; validation, S.A.B.; formal analysis, J.A.; investigation, S.A.B.; data curation, S.A.B.; writing—original draft preparation, S.A.B.; writing—review and editing, J.A.; supervision, J.A.; project administration, J.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset is publicly archived and available at <https://www.variotdbs.pl/> (accessed on 15 August 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
CVSS	Common Vulnerability Scoring System
NVD	National Vulnerability Database
DL	Deep Learning
CNN	Convolutional neural network
CVE	Common Vulnerabilities and Exposures
BERT	Bidirectional Encoder Representations from Transformers.
ERR	Error Reduction Rate

References

1. Vailshery, L. Number of iot Connections Worldwide 2022–2033, with Forecasts to 2030. 2024. Available online: <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/> (accessed on 17 October 2025).
2. Abawajy, J.; Huda, S.; Sharmeen, S.; Hassan, M.M.; Almogren, A. Identifying cyber threats to mobile-IoT applications in edge computing paradigm. *Future Gener. Comput. Syst.* **2018**, *89*, 525–538. [\[CrossRef\]](#)
3. AlJabri, Z.; Abawajy, J.; Huda, S. MDS-Based Cloned Device Detection in IoT-Fog Network. *IEEE Internet Things J.* **2024**, *11*, 22128–22139. [\[CrossRef\]](#)
4. Baho, S.A.; Abawajy, J. Analysis of Consumer IoT Device Vulnerability Quantification Frameworks. *Electronics* **2023**, *12*, 1176. [\[CrossRef\]](#)
5. Shahidinejad, A.; Abawajy, J. An all-inclusive taxonomy and critical review of blockchain-assisted authentication and session key generation protocols for IoT. *ACM Comput. Surv.* **2024**, *56*, 186. [\[CrossRef\]](#)
6. Spring, J.; Hatleback, E.; Householder, A.; Manion, A.; Shick, D. Time to Change the CVSS? *IEEE Secur. Priv.* **2021**, *19*, 74–78. [\[CrossRef\]](#)
7. Malhotra, R.; Vidushi. Severity prediction of software vulnerabilities using textual data. In *Proceedings of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications (ICMISC)*; Springer: Singapore, 2020; pp. 453–464.
8. Saklani, S.; Kalia, A. Severity prediction of software vulnerabilities using convolutional neural networks. *Inf. Comput. Secur.* **2025**, *33*, 613–630. [\[CrossRef\]](#)
9. Ni, X.; Zheng, J.; Guo, Y.; Jin, X.; Li, L. Predicting severity of software vulnerability based on BERT-CNN. In *Proceedings of the International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, Shijiazhuang, China, 22–24 July 2022; IEEE: New York, NY, USA, 2022; pp. 711–715.
10. Shahid, M.R.; Debar, H. Cvss-bert: Explainable natural language processing to determine the severity of a computer security vulnerability from its description. In *Proceedings of the 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Virtually Online, 13–15 December 2021; pp. 1600–1607.
11. Costa, J.C.; Roxo, T.; Sequeiros, J.B.; Proenca, H.; Inacio, P.R. Predicting CVSS metric via description interpretation. *IEEE Access* **2022**, *10*, 59125–59134. [\[CrossRef\]](#)
12. Marali, M.; Balakrishnan, K. Vulnerability Classification Based on Fine-Tuned BERT and Deep Neural Network Approaches. In *Proceedings of the International Conference on Intelligent Systems and Sustainable Computing*, Hyderabad, India, 16–17 December 2022; Springer: Singapore, 2022; pp. 257–268.
13. Sanh, V. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
14. Nurse, J.R.; Creese, S.; De Roure, D. Security risk assessment in internet of things systems. *IT Prof.* **2017**, *19*, 20–26. [\[CrossRef\]](#)
15. Janiszewski, M.; Rytel, M.; Lewandowski, P.; Romanowski, H. VARIOt-Vulnerability and Attack Repository for the Internet of Things. In *Proceedings of the 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, Taormina, Italy, 16–19 May 2022; IEEE: New York, NY, USA, 2022; pp. 752–755.
16. Zhang, Z.; Kumar, V.; Mayo, M.; Bifet, A. Assessing Vulnerability from Its Description. In *Proceedings of the International Conference on Ubiquitous Security*, Zhangjiajie, China, 28–31 December 2022; IEEE: New York, NY, USA, 2022; pp. 129–143.
17. Jiang, Y.; Atif, Y. An approach to discover and assess vulnerability severity automatically in cyber-physical systems. In *Proceedings of the 13th International Conference on Security of Information and Networks*, Merkez, Turkey, 4–7 November 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1–8.
18. Shan, C.; Zhang, Z.; Zhou, S. A multi-task deep learning based vulnerability severity prediction method. In *Proceedings of the 2023 IEEE 12th International Conference on Cloud Networking (CloudNet)*, Hoboken, NJ, USA, 1–3 November 2023; IEEE: New York, NY, USA, 2023; pp. 307–315.
19. Wang, Y.; Zhang, J.; Huang, M. LNet: Adaptive Latent Space Networks for Vulnerability Severity Assessment. *Information* **2025**, *16*, 779. [\[CrossRef\]](#)

20. Zhang, Z.; Kumar, V.; Pfahringer, B.; Bifet, A. Ai-enabled automated common vulnerability scoring from common vulnerabilities and exposures descriptions. *Int. J. Inf. Secur.* **2025**, *24*, 16.
21. Mirtaheri, S.L.; Pugliese, A.; Movahedkor, N.; Majd, A. Advanced automated vulnerability scoring: Improving performance with a fine-tuned BERT-CNN model. In Proceedings of the 11th International Symposium on Telecommunications (IST), Sofia, Bulgaria, 21–22 November 2022; IEEE: New York, NY, USA, 2024; pp. 109–113.
22. Aghaei, E.; Al-Shaer, E.; Shadid, W.; Niu, X. Automated CVE analysis for threat prioritization and impact prediction. *arXiv* **2023**, arXiv:2309.03040. [[CrossRef](#)]
23. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
24. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. 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, MN, USA, 2–7 July 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1 (Long and Short Papers), pp. 4171–4186.
25. Jain, S.M. *Introduction to Transformers for NLP: With the Hugging Face Library and Models to Solve Problems*; Apress: Berkeley, CA, USA, 2022.
26. *NISTIR-7946*; CVSS Implementation Guidance. National Institute of Standards and Technology: Gaithersburg, MD, USA, 2014.
27. Bates, S.; Hastie, T.; Tibshirani, R. Cross-validation: What does it estimate and how well does it do it? *J. Am. Stat. Assoc.* **2024**, *119*, 1434–1445. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.