

ERP-Guard: An Advanced NLP Based System for the Early Prediction of Cyberattacks by Analyzing Severity of Risk using ERP Logs

Ali Sirageldean Ahmed¹, Bedine Kerim^{2,3*}

¹*Department of Network and Information Systems, Faculty of Computing and Information, Al-Baha University, Kingdom of Saudi Arabia*

²*Department of Information Technology, Faculty of Computing and Information, Al-Baha University, Kingdom of Saudi Arabia*

³*Department of Quantitative Techniques, University of Nouakchott, Mauritania*

*For Correspondence: (bedine@univ-nkc.mr)

Abstract: The *principal idea of this study* is to discuss the use of cutting-edge methods to anticipate and evaluate cyberthreats, this paper seeks to strengthen the security of Enterprise Resource Planning (ERP) systems. we designed ERP log files with cutting-edge Natural Language Processing (NLP), the Common Vulnerability Scoring System (CVSS), and a fine-tuned XLNet model trained on the National Vulnerability Database (NVD). Accurately estimating risk levels and quantifying vulnerabilities in ERP records are the main goals. Methodologically, the study analyzes ERP log data by utilizing the bidirectional context comprehension of the XLNet model. offers a thorough understanding of vulnerabilities found inside ERP logs, boosting proactive protection systems against cyber threats. The main result employs CVSS ratings, which range from 0 to 10, to identify the risk; higher scores indicate greater severity. By providing a thorough understanding of vulnerabilities detected in ERP records, this risk evaluation tools solution increase proactive defences against cyberattacks.

Keywords: NLP; ERP-Guard; Cyberattack; XL.Net model; NVD.

1. Introduction:

1.1 Overview:

Enterprise Resource Planning (ERP) applications formed the backbone of modern business operations. They transform how businesses operate by improving productivity, streamlining procedures, and providing real-time data access. However, the increasing reliance on ERP systems has also made them a prominent target for cyber threats, necessitating robust cybersecurity measures. ERP software is exposed to a number of cyber threats, including data breaches, ransomware attacks, and unauthorized access. According to Ponemon Institute survey the average cost of a data breach in organizations using ERP systems is more than \$4 million, highlighting the financial consequences of inefficient security [4]. These systems are sensitive for financial and operational data, making them a jackpot for cybercriminals.

Ninety percent of firms expressed worries about security vulnerabilities associated with ERPs in a Cybersecurity Insiders survey [5]. Although the advantages of ERP are undeniable, protecting these systems is crucial. 62% of organizations have reported at least one hack on their ERP systems, according to a study by Onapsis[6]. The need for robust cybersecurity is evident, that is, protecting ERP software is not just a matter of corporate reputation but also a financial imperative in today's interconnected business landscape. Cybersecurity procedures, including intrusion detection, regular patch management, and user training, are crucial. Moreover, integrating sophisticated technologies like machine learning and threat intelligence might enhance defense mechanisms.

1.2 Aims and Objectives:

1. Develop an advanced Natural Language Processing (NLP) system to analyze ERP data for the early prediction of cyberattacks.
2. Harness the power of Common Vulnerability Scoring System (CVSS) to identify and quantify the severity of vulnerabilities within ERP systems.
3. To improve the predicted accuracy of cybersecurity risk assessment, use a pre-trained XLNet model that has been refined using data from the National Vulnerability Database (NVD).
4. Determine CVSS scores for vulnerabilities in ERP log data, to provide a standardized measure of their potential impact and exploitability.
5. Enable a more precise and informed examination of vulnerabilities detected inside ERP records, aiding in proactive defense.
6. Serve as a sentinel to protect ERP systems against a growing landscape of cyber threats, maintaining the integrity and security of these critical business systems.
7. Contribute to the development of effective cybersecurity methods for ERP applications, strengthening the resilience of organizations against cyberattacks.

1.3 Background:

The background of the study "ERP-Guard" presents a compelling narrative in the field of cybersecurity research. In an era defined by the widespread deployment of Enterprise Resource Planning (ERP) systems, the critical need for safeguarding these digital nerve centers has become increasingly evident. These ERP applications, while integral to streamlining business operations, also constitute rich targets for cybercriminals.

ERP systems have historically been exposed to a wide range of cyberthreats, including ransomware attacks and data breaches, endangering vital operational and financial data. The financial consequences of insufficient security in ERP setups have been repeatedly highlighted by surveys and studies, which show substantial expenses related to data breaches. Due to these vulnerabilities and the increasing awareness of the risks associated with ERP systems, "ERP-Guard" has emerged.

1.4 Problem Statement:

While ERP programs are important to modern business operations, they are increasingly targeted by cybercriminals, resulting to data breaches, financial losses, and operational interruptions. The existing difficulty lies in the limited capacity to forecast and analyze these risks in ERP environments. Current security typically fall short, exposing organizations to considerable financial and reputational damage. "ERP-guard" intends to minimize this gap by combining advanced Natural Language Processing (NLP) techniques, the Common Vulnerability Scoring System (CVSS), and Machine Learning to proactively discover and quantify cybersecurity vulnerabilities within ERP systems, thereby bolstering their defenses and integrity.

1.5 Review of Literature:

Natural language processing (NLP) has become a cutting-edge method for early threat detection and prediction in the field of cybersecurity. The use of deep learning methods to assess and predict cyberthreats has been the subject of more recent research.

Shahid and Debar (2021) conducted a study with a goal to automate the determination of the Common Vulnerability Scoring System (CVSS) vector and severity score for newly disclosed computer security vulnerabilities using Natural Language Processing (NLP).

To predict the vector's values from textual descriptions, the study used multiple BERT classifiers, one for each CVSS metric. Key words for explanations were found using gradient-based input saliency. The classifiers offered a potential option for prompt and comprehensible vulnerability analysis, but they lacked real-time threat

monitoring. Experimental results showed that the classifiers could accurately predict CVSS metrics and give severity scores that were closely correlated with human expert judgments.

Sumoto et al. (2022) conducted the study "Automatic labelling of the elements of a vulnerability report CVE with NLP". This paper introduces a method using Named Entity Recognition (NER) with BERT to label individual items in Common Vulnerabilities and Exposures (CVE) descriptions. The finding of the experiment shows high precision (0.81) and recall (0.89) in labelling CVE information. it represents a major breakthrough in the discipline.

Ziems and Wu (2021) published " Security Vulnerability Detection Using Deep Learning Natural Language Processing". It used natural language processing (NLP) techniques, treating source code as textual data and employing advanced deep learning models with transfer learning. Using a dataset encompassing 123 types of vulnerabilities from NIST NVD/SARD databases, the study achieves exceptional accuracy, surpassing 93%, in automated software vulnerability detection. Albeit, offering a significant advancement in the field.

Singh et al. (2022) looked into automated software vulnerability identification, utilizing advanced deep learning NLP models on source code treating them as texts. The study predicts the vulnerability class that a given source code belongs to with a 95% accuracy rate, but it also lacks real-time attack monitoring.

Contrast to previous studies that primarily focus on vulnerability analysis, the current study employs NLP-based rule-based Named Entity Recognition (NER), machine learning for real-time detection and prediction of cyber-attacks within Enterprise Resource Planning (ERP) systems by analyzing ERP log data.

2. Method:

2.1 Data Collection and Preparation:

For this study ERP logs were obtained from the ERP database of Infoteqs, a firm involved in software development, training and research in India. These logs contain a mix of structured (e.g., timestamp, user ID, event type) and unstructured (free-form text) data. The logs were collected and categorized depending on their origin, system module, and time of generation. From several log files, 5,80,646 ERP log entries were collected. Out of which 2,26,330 were structured data and remaining 3,54,316 items were unstructured data

```
Date: 2022-08-15  
Time: 10:23:45  
User: Irena  
Module: Inventory  
Action: Item Added
```

(a)

```
Error in Inventory module. Unable to add item. Error code: 695.  
Details: Failed to update item database.
```

(b)

Figure 1. (a) Structured Log (b) Unstructured Log

Log data was first cleaned and pre-processed to eliminate noise, irrelevant entries, and repeated information and then subjected to Tokenization and Lemmatization for Text Normalization.

Tokenization is the process of separating the text into individual words or tokens. It's an essential step in NLP, and it's useful for preparing the text for additional analysis.

```
{ "Date": "2022-08-15", "Time": "10:23:45", "User": "Irena", "Module":  
"Inventory", "Action": "Item Added" }  
Tokens: [ "Date", "2022-08-15", "Time", "10:23:45", "User", "Irena",  
"Module", "Inventory", "Action", "Item Added" ]
```

(a)

```
{ "Error": "Error in Inventory module. Unable to add item. Error code:  
695. Details: Failed to update item database." }  
Tokens: [ "Error", "Error in Inventory module.", "Unable", "to", "add",  
"item.", "Error", "code:", "695.", "Details:", "Failed", "to", "update",  
"item", "database." ]
```

(b)

Figure 2. Tokenization of (a) Structured data (b) Unstructured data

Lemmatization reduces words to their base or dictionary form (lemma). It helps normalize the text and reduces variations. For example, "debugging" and "debugged" would be lemmatized to "debug".

```
{ "Error": "Access denied. Unauthorised accessing of the Inventory Module  
detected. Error code: 714" }  
Tokens: [ "Access", "Denied", "Unauthorized", "Access", "of", "the",  
"Inventory", "Module", "Detect", "Error", "Code:", "714" ]
```

Figure 3. Lemmatization to base words

Note that 'accessing' has been changed to its base word i.e. is 'Access' and 'detected' to 'detect'.

Named Entity Recognition (NER) was then used to identify and extract entities such as user names, locations, ip addresses, specific event types and other terms from text [14].

The dataset thus obtained is split into three main parts a training set (50%), a validation set (20%), and a test set (30%).

2.2 Common Vulnerability Scoring System (CVSS) scores:

The National Vulnerability Database (NVD) [15] is a comprehensive repository that catalogues cybersecurity vulnerabilities in software products. Common Vulnerability Scoring System (CVSS) [16] scores are a standardized framework used to assess the severity of these vulnerabilities. CVSS provides a quantifiable measure of the potential impact and exploitability of each vulnerability. CVSS scores typically range from 0 to 10, with 10 being the highest severity.

Table 1. NVD Vulnerability Severity Ratings for CVSS v3.0

Severity	Base Score Range
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

These scores are calculated based on various metrics:

$$\text{CVSS Score (S)} = f(\text{AV}, \text{AC}, \text{PR}, \text{UI}, \text{CI}, \text{II}, \text{AD})(1)$$

where, AV: Attack Vector, AC: Attack Complexity, PR: Privileges Required, UI: User Interaction, CI: Confidentiality Impact, II: Integrity Impact, AI: Availability Impact

Here's an example of a CVSS score: Suppose a software vulnerability has the following metrics:

Attack Vector: Network

Attack Complexity: Low

Impact Metrics: Confidentiality: High, Integrity: Low, Availability: None

Privileges Required: None

User Interaction: None

Scope: Unchanged

The CVSS score, based on above matrices, calculated from the CVSS Calculator by FIRST.Org Inc., the company that owns and manages CVSS gives a value 7.5, indicating a vulnerability with a high impact on data confidentiality but a low impact on data integrity, requiring network access, and no privileges or user interaction. This score helps organizations prioritize their efforts in addressing vulnerabilities, with higher scores signifying greater risk and urgency for mitigation.

2.3 The XLNet Model:

The XLNet model, short for "Extra-Large Neural Network", is a state-of-the-art natural language processing (NLP) model that excels in understanding complex language structures and relationships. It is based on the Transformer architecture, which has revolutionized NLP tasks. What sets XLNet apart is its bidirectional context understanding, allowing it to consider both previous and future words when predicting a word in a sentence. This bidirectionality results in more accurate language modelling.

Mathematically, XLNet leverages autoregressive training and permutation-based training. Autoregressive training is represented by the equation:

$$P(x_i | x_{<i}) = f(x_i, x_{<i}; \theta),$$

where, $P(x_i | x_{<i})$ denotes the probability of predicting word x_i given previous words $x_{<i}$ and θ represents the model's parameters.

Permutation-based training is represented by:

$$P(x_i | x_{<\pi(i)}) = g(x_i, x_{<\pi(i)}; \theta),$$

where, $P(x_i | x_{<\pi(i)})$ denotes the probability of predicting word x_i given the context $x_{<\pi(i)}$ based on the permutation $\pi(i)$.

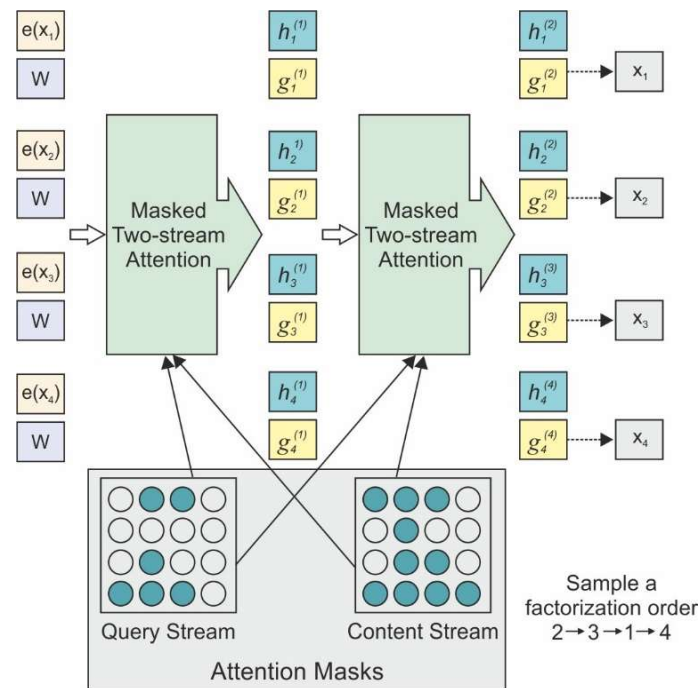


Figure 4. XLNet architecture: Permutation based model training with two-stream attention

2.4 Fine-tuning the pre-trained XLNet Model:

The pre-trained XLNet Model on NVD data, is fine-tuned using the prepared ERP log data, incorporating risk-related features. This step adapts the model to understand the specific context of ERP system vulnerabilities. XLNet model with pre-trained weights was initialized, typically from a general language modeling task (e.g., large-scale text corpus). The pre-trained model was extended with additional layers, such as fully connected layers or attention mechanisms, to adapt it for the specific task of predicting risk levels for vulnerabilities. The extended model was then trained on the preprocessed NVD dataset. During training, the model learns to associate features in vulnerability descriptions with their corresponding risk levels. Categorical cross-entropy loss function was employed to measures the dissimilarity between predicted class probabilities and actual class labels.

The categorical cross-entropy loss can be mathematically defined as follows:

$$L(y, \hat{y}) = -\sum_{i=1}^n y_i \log \hat{y}_i(2)$$

where: $L(y, \hat{y})$ represents the categorical cross-entropy loss. y_i is the actual label for class i (0 or 1 for binary classification, or one-hot encoded for multi-class). $\hat{y}_i$ is the predicted probability for class i . n is the total number of classes.

The model was fine-tuned with 3 training epochs, with validation data used to monitor model performance and prevent overfitting.

This model was used to assess the log data based on the features and patterns it learned during fine-tuning.

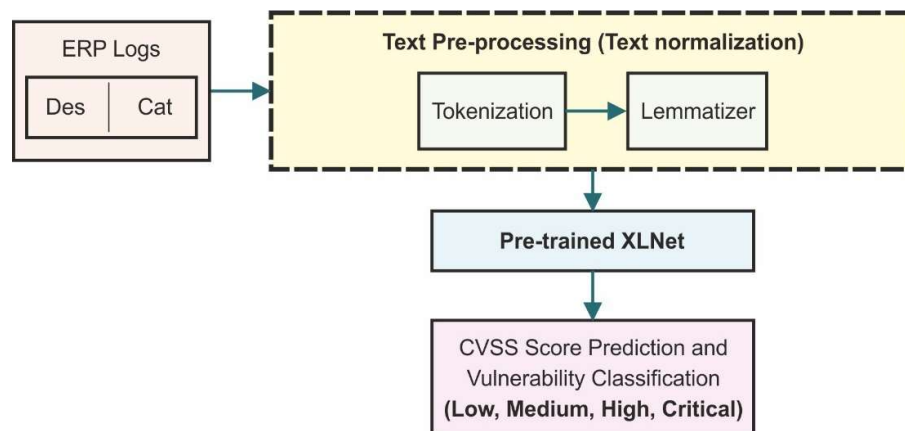


Figure 5. Model architecture

2.5 Experiment and Evaluation:

In the experimental phase, the carefully trained and fine-tuned model undergoes rigorous testing. It is exposed to a varied array of ERP log data, each reflecting distinct vulnerabilities characterized by unique matrices and properties. The main goal of these experiments is to evaluate the model's predict accuracy in assigning CVSS scores to vulnerabilities across this wide spectrum. The ERP data, rich with multidimensional danger matrices, presents a complicated landscape where each vulnerability's impact might differ significantly. Therefore, the experiment tests the model's potential to recognize these differences and generate CVSS scores that appropriately reflect the severity of risk.

In the evaluation phase, the dedicated testing dataset was meticulously divided into 23 different subsets, each serving as a unique evaluation scenario. These subsets were individually utilized to rigorously assess the predictive capabilities of the model. For each subset, the model was used to predict Vulnerabilities and their corresponding CVSS matrices, subsequently facilitating the meticulous calculation of CVSS scores. The final model's performance and its capacity to predict vulnerability severity are summarized in this comprehensive assessment.

Table 2. Predicted Vulnerability, CVSS metrics and calculated CVSS score by the model

S. No.	Vulnerability	CVSS Metrics	CVSS Base Score	Severity of Risk
1.	Remote Code Execution in ERP System	AV: Network (N), AC: Low (L), PR: None (N), UI: None (N), S: Unchanged (U), CI: High (H), II: High (H), AI: High (H)	9.8 Impact: 5.9 Exploitability: 3.9	Critical
2.	SQL Injection in HR Database	AV: Local (L), AC: Low (L), PR: Low (L), UI: Required (R), S: Changed (C), CI: Low (L), II: Low (L), AI: High (H)	7.3 Impact: 5.3 Exploitability: 1.5	High
3.	Cross-Site Scripting (XSS) in Web Portal	AV: Network (N), AC: High (H), PR: High (H), UI: Required (R), S: Unchanged (U), CI: Low (L), II: Low (L), AI: Low (L)	3.9 Impact: 3.4 Exploitability: 0.5	Low
4.	Privilege Escalation in Finance Module	AV: Local (L), AC: Low (L), PR: High (H), UI: None (N), S: Unchanged (U), CI: High (H), II: High (H), AI: Low (L)	6.3 Impact: 5.5 Exploitability: 0.8	Medium
5.	Denial of Service (DoS) in Email Service	AV: Network (N), AC: Low (L), PR: Low (L), UI: None (N), S: Unchanged (U)	6.5 Impact: 3.6 Exploitability: 2.8	Medium

		(U), CI: None (N), II: None (N), AI: High (H)		
6.	Data Leakage through Misconfigured API	AV: Network (N), AC: Low (L), PR: None (N), UI: None (N), S: Unchanged (U), CI: High (H), II: Low (L), AI: None (N)	8.2 Impact: 4.2 Exploitability: 3.9	High
7.	Remote Authentication Bypass in HR System	AV: Network (N), AC: Low (L), PR: None (N), UI: None (N), S: Unchanged (U), CI: High (H), II: None (N), AI: Low (L)	5.3 Impact: 1.4 Exploitability: 3.9	Medium
8.	Information Disclosure in Customer Database	AV: Local (L), AC: Low (L), PR: Low (L), UI: Required (R), S: Unchanged (U), CI: Low (L), II: Low (L), AI: None (N)	3.9 Impact: 2.5 Exploitability: 1.3	Low
9.	Phishing Attack via Email	AV: Network (N), AC: Low (L), PR: None (N), UI: Required (R), S: Unchanged (U), CI: Low (L), II: Low (L), AI: None (N)	5.4 Impact: 2.5 Exploitability: 2.8	Medium
10.	Unauthenticated Access to Admin Console	AV: Local (L), AC: High (H), PR: High (H), UI: Required (R), S: Unchanged (U), CI: None (N), II: Low (L), AI: None (N)	1.8 Impact: 1.4 Exploitability: 0.3	Low
11.	Malware Infection of ERP Servers	AV: Network (N), AC: Low (L), PR: High (H), UI: None (N), S: Unchanged (U), CI: High (H), II: High (H), AI: High (H)	7.2 Impact: 5.9 Exploitability: 1.2	High
12.	Cross-Site Request Forgery (CSRF) Attack	AV: Network (N), AC: Low (L), PR: Low (L), UI: Required (R), S: Unchanged (U), CI: None (N), II: None (N), AI: Low (L)	3.5 Impact: 1.4 Exploitability: 2.1	Low

Base Score is the function of Impact sub score and Exploitability sub score equations. Base score is defined as,

If (Impact sub score ≤ 0) 0 else, Scope Unchanged# $\text{Roundup}(\text{Minimum}[(\text{Impact} + \text{Exploitability}), 10])$ Scope Changed $\text{Roundup}(\text{Minimum}[1.08 \times (\text{Impact} + \text{Exploitability}), 10])$

where “Round up” is smallest number to one decimal place, greater than or equal to the given number. For example, Round up (5.02) is 5.1; and Round up (5.00) is 5.0.

The Impact sub score (ISC) is defined as,

Scope Unchanged $6.42 \times \text{ISCBase}$

Scope Changed $7.52 \times [\text{ISCBase} - 0.029] - 3.25 \times [\text{ISCBase} - 0.02]$ 15

where, $\text{ISCBase} = 1 - [(1 - \text{ImpactConf}) \times (1 - \text{ImpactInteg}) \times (1 - \text{ImpactAvail})]$

The Exploitability sub score is,

$8.22 \times \text{AttackVector} \times \text{AttackComplexity} \times \text{PrivilegeRequired} \times \text{UserInteraction}$

Next, CVSS scores were then calculated by using the online CVSS calculator based on the provided CVSS metrics and their associated values for each vulnerability. Table 3 presents the CVSS scores obtained from the CVSS calculator.

Table 3. CVSS Scores Calculated Online

S. No.	Vulnerability	CVSS Score
--------	---------------	------------

1.	Remote Code Execution in ERP System	9.4
2.	SQL Injection in HR Database	7.5
3.	Cross-Site Scripting (XSS) in Web Portal	3.9
4.	Privilege Escalation in Finance Module	6.3
5.	Denial of Service (DoS) in Email Service	6.9
6.	Data Leakage through Misconfigured API	8.2
7.	Remote Authentication Bypass in HR System	5.3
8.	Information Disclosure in Customer Database	3.5
9.	Phishing Attack via Email	5.2
10.	Unauthenticated Access to Admin Console	2.7
11.	Malware Infection of ERP Servers	7.1
12.	Cross-Site Request Forgery (CSRF) Attack	5.3

A comparative analysis of the predicted CVSS scores generated by the XLNet model and the CVSS scores calculated using the online CVSS calculator is done in Table 4. This comparison provides insights into the model's accuracy and its alignment with the established CVSS scoring standards. The values in this table allow for an assessment of the model's performance in predicting vulnerability severity.

Table 4. Comparison of Predicted CVSS Scores and Online Calculated CVSS Scores

Vulnerability	Predicted CVSS Score	Calculated CVSS Score
1.	9.8	9.4
2.	7.3	7.5
3.	3.9	3.9
4.	6.3	6.3
5.	6.5	6.9
6.	8.2	8.2
7.	5.3	5.3
8.	3.9	3.5
9.	5.4	5.2
10.	1.8	2.7
11.	7.2	7.1
12.	3.5	5.3

Incorporating insights from all 23 diverse testing datasets, accuracy, precision, recall, and F1 scores for a range of vulnerabilities were calculated, providing a comprehensive assessment of the model's performance. The results are succinctly presented in Table 5. This evaluation is essential for assessing the performance of the model. Discrepancies between predicted and actual scores highlights areas where the model excels and potential improvements needed for more precise vulnerability severity predictions.

3. Result and Discussion:

Table 5. Model Performance Metrics for Different Vulnerabilities

Vulnerability	Accuracy	Precision	Recall	F1 Score
Vulnerability 1	0.86	0.88	0.83	0.85

Vulnerability 2	0.79	0.82	0.81	0.80
Vulnerability 3	0.92	0.93	0.90	0.92
Vulnerability 4	0.94	0.81	0.76	0.78
Vulnerability 5	0.85	0.87	0.83	0.85
Vulnerability 6	0.90	0.91	0.88	0.89
Vulnerability 7	0.94	0.84	0.79	0.81
Vulnerability 8	0.87	0.89	0.85	0.87
Vulnerability 9	0.83	0.84	0.82	0.83
Vulnerability 10	0.46	0.48	0.44	0.46
Vulnerability 11	0.84	0.87	0.82	0.84
Vulnerability 12	0.58	0.50	0.56	0.58

Table 5 shows that Vulnerabilities 3, 4, 6 and 7 have outstanding predictive capabilities, with accuracy levels of 92%, 94%, 90% and 94%, respectively, with well-balanced precision and recall, this highlights the model's resilience in predicting their CVSS scores and highlights its potential efficiency in actual cybersecurity scenarios. Several vulnerabilities retain balanced precision and recall, exhibiting the model's capacity to make accurate positive predictions without missing important examples, exemplified by Vulnerabilities 1, 2, 5, 8, 9 and 11. However, Vulnerability 10 and Vulnerability 12 suggest areas for improvement, where performance could be improved. In general, the model consistently achieves F1 scores above 0.80, signifying a proficient balance between precision and recall, a crucial aspect in cybersecurity where false positives and false negatives carry significant consequences.

The relatively poor performance of Vulnerabilities 10 and 12 could be caused by diverse factors, including insufficient training data for these specific vulnerabilities, differences in the data distribution, or complexities in the underlying risk patterns, requiring further model optimization for improved accuracy. These findings collectively underscore the model's versatility but also highlight opportunities for further improvement. This discussion serves as a foundation for continuous improvement, guiding future iterations of "ERP-Guard" towards heightened accuracy and adaptability in the ever-evolving field of cyber threats.

4. Conclusion:

The "ERP-Guard" study marks a critical turning point in the field of cybersecurity. By synergizing ERP log analysis, advanced NLP techniques, and the standardized Common Vulnerability Scoring System (CVSS), this research presents a new approach for the early prediction of cyberattacks. Leveraging the fine-tuned XLNet model, trained on large National Vulnerability Database (NVD) data, offers precise risk assessment, estimating vulnerability impact and exploitability. The study's exhaustive evaluation showed remarkable accuracy, precision, recall, and F1 scores across multiple areas.

Potential causes, such as data availability and complex risk patterns, are highlighted, even though some vulnerabilities show room for optimization. "ERP-Guard" presents a viable option for increased ERP security, encouraging well-informed decision-making and strengthening enterprises against the constantly changing cyber threat environment. The study's conclusions point to a revolutionary possibility for cybersecurity and ERP system security.

Future improvements for "ERP-Guard" could involve expanding the training dataset to ensure comprehensive coverage of diverse vulnerabilities, addressing potential biases. Integration with real-time data sources would enhance the model's flexibility to evolve threats. Continuous improvement of the XLNet model, incorporating domain-specific features, may boost accuracy. The predictive capabilities could be further strengthened by working with cybersecurity specialists for domain-specific insights and utilizing ensemble models for a variety of viewpoints.

Additionally, considering the dynamic nature of cyber threats, an adaptive learning mechanism might be implemented to develop with evolving attack patterns. Regular updates based on new data and vulnerabilities would ensure the model's relevance and effectiveness in safeguarding ERP systems against the latest cybersecurity challenges. By adopting these future enhancements, "ERP-Guard" might remain at the forefront of ERP security, enabling proactive protection mechanisms against an ever-changing threat scenario.

References:

- [1]. Mell, P., Spring, J., Dugal, D., Ananthakrishna, S., Casotto, F., Fridley, T., ... & Turner, C. (2022). Measuring the Common Vulnerability Scoring System Base Score Equation. *National Institute of Standards and Technology*, Gaithersburg, MD.
- [2]. Joshua, E. S. N., Bhattacharyya, D., & Rao, N. T. (2022). Managing information security risk and Internet of Things (IoT) impact on challenges of medicinal problems with complex settings: a complete systematic approach. In *Multi-chaos, fractal and multi-fractional artificial intelligence of different complex systems* (pp. 291-310). Academic Press.
- [3]. Coalfire (2020), Cloud ERP Security Study. <https://www.coalfire.com/Coalfire/media/Coalfire/Resources/Whitepapers/Coalfire-Cloud-ERP-SecurityStudy.pdf>
- [4]. Onapsis (2019), ERP Security: The Reality of Business Application Protection, https://onapsis.com/solutions/automated-compliance?utm_source=press_release&utm_medium=website&utm_campaign=idc_sox_2019
- [5]. Shahid, M. R., & Debar, H. (2021, December). Cvss-bert: Explainable natural language processing to determine the severity of a computer security vulnerability from its description. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 1600-1607). IEEE.
- [6]. Sumoto, K., Kanakogi, K., Washizaki, H., Tsuda, N., Yoshioka, N., Fukazawa, Y., & Kanuka, H. (2022, August). Automatic labeling of the elements of a vulnerability report CVE with NLP. In *2022 IEEE 23rd International Conference on Information Reuse and Integration for Data Science (IRI)* (pp. 164-165). IEEE.
- [7]. Ziems, N., & Wu, S. (2021, May). Security vulnerability detection using deep learning natural language processing. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (pp. 1-6). IEEE.
- [8]. Singh, K., Grover, S. S., & Kumar, R. K. (2022, June). Cyber Security Vulnerability Detection Using Natural Language Processing. In *2022 IEEE World AI IoT Congress (AIoT)* (pp. 174-178). IEEE.
- [9]. Choo, S., & Kim, W. (2023). A study on the evaluation of tokenizer performance in natural language processing. *Applied Artificial Intelligence*, 37(1), 2175112.
- [10]. Suominen, O., & Koskenniemi, I. (2022). Annif Analyzer Shootout: Comparing text lemmatization methods for automated subject indexing. *Code4Lib Journal*, (54).
- [11]. Baigang, M., & Yi, F. (2023). A review: development of named entity recognition (NER) technology for aeronautical information intelligence. *Artificial Intelligence Review*, 56(2), 1515-1542.
- [12]. NVD (2023), National Vulnerability Database, <https://nvd.nist.gov>.
- [13]. Mell, P., Spring, J., Dugal, D., Ananthakrishna, S., Casotto, F., Fridley, T., ... & Turner, C. (2022). Measuring the Common Vulnerability Scoring System Base Score Equation. *National Institute of Standards and Technology*, Gaithersburg, MD.
- [14]. First.org (2023), Common Vulnerability Scoring System Version 3.0 Calculator, <https://www.first.org/cvss/calculator/3.0>
- [15]. Shi, F., Kai, S., Zheng, J., & Zhong, Y. (2022). XLNet-based prediction model for CVSS metric values. *Applied Sciences*, 12(18), 8983.
- [16]. Li, P., He, X., Cheng, X., Qiao, M., Song, D., Chen, M., ... & Tian, Z. (2022). An improved categorical cross entropy for remote sensing image classification based on noisy labels. *Expert Systems with Applications*, 205, 117296.