Journal of Technology Research (JTR)
Volume 4, Issue 1, (2026), pp 051-062, ISSN 3005-639X

# Evaluating The Impact Of Time-To-Exploit Estimation For Vulnerability Prioritization

Amira Khalifa Ellabad [*1], Juma Ibrahim [1]

[1] Postgraduate Office, Software Development Technology, College of Computer Technology Tripoli (CCTT), Tripoli, Libya,

*Corresponding author email: al2303002@cctt.edu.ly

Received: 15-12-2025 | Accepted: 14-04-2026 | Available online: 23-04-2025 | DOI:10.26629/jtr.2025.**

**ABSTRACT**

Recently, security vulnerabilities have increased significantly, as this study addresses the issue of prioritizing them by developing a predictive model that estimates the time required to exploit them. Data obtained from multiple sources was used to develop this model, including a unified Kaggle dataset, which combines data from three reliable sources: the National Vulnerability Database (NVD), the CISA Known Exploitable Vulnerabilities (KEV) list, and the Exploitation Prediction Score System (EPSS). Data from both ExploitDB and CISA KEV list was also used. The dataset was divided into training (2021-2023) and testing (2024) sets, to compensate for the lack of confirmed exploitation dates, isotonic regression was used to model the monotonic relationship between EPSS scores and actual exploitation dates, as a methodological alternative. We also evaluated three regression models: the best results for the test set were shown in the XGBoost model (MAE=2.98 days, RMSE=12.20 days, $R^2$=0.936, MAPE=14.43%), while the Random Forest performed the baseline linear regression model (MAE=2.77, RMSE=14.59, $R^2$=0.908, MAPE=13.43% vs. MAE=18.48, RMSE=24.57, $R^2$=0.740, MAPE=51.50%). To interpret these predictions into actionable information, the estimated "Time To Exploit" was transformed into a "Composite Priority Index" that combines the predicted speed of exploitation with the probability score, the Exploitation Potential Scoring System (EPSS) was then used to categorize vulnerabilities into the following levels: urgent, high, medium, and low. This approach improved our ability to identify high-risk vulnerabilities early by incorporating time-based data, compared to relying solely on static criteria.The results show that incorporating the time dimension enhances its reliability and wider applicability.

**Keywords:** Cybersecurity, Vulnerability Prioritization, Time-To-Exploit (TTE), Exploit Prediction Scoring System (EPSS), Regression Models.

# تقييم تأثير تقدير وقت الاستغلال لتحديد أولويات الثغرات الأمنية

أميرة خليفة اللباد[1]، جمعة إبراهيم 1

[1,2] مكتب الدراسات العليا ، قسم هندسة تطوير البرمجيات ، كلية تقنية الحاسوب طرابلس ، طرابلس، ليبيا

**ملخـــــــص البحـــــــث**

في الآونة الاخيرة ازدادت الثغرات الامنية بشكل كبير، حيث تتناول هذه الدراسة مسألة تحديد أولوياتها من خلال تطوير نموذج تنبؤي يقدر الوقت اللازم لاستغلالها. تم استخدام البيانات التي تم الحصول عليها من مصادر متعددة لتطوير هذا النموذج، بما في ذلك

مجموعة بيانات Kaggle موحدة، والتي تجمع البيانات من ثلاثة مصادر موثوقة: قاعدة بيانات الثغرات الوطنية (NVD)، وقائمة الثغرات المعروفة القابلة للاستغلال (KEV) التابعة لوكالة الأمن السيبراني (CISA)، ونظام نقاط التنبؤ بالاستغلال (EPSS)، كما تم استخدام بيانات كلا من ExploitDB وقائمة CISA KEV ، وتم تقسيم مجموعة البيانات إلى مجموعتين: مجموعة للتدريب (2021–2023) ومجموعة للاختبار (2024)، للتعويض عن عدم وجود تواريخ استغلال مؤكدة، تم استخدام الانحدار المتساوي التوتر لنمذجة العلاقة الرتيبة بين درجات EPSS وتواريخ الاستغلال الفعلية كبديل منهجي، كما قمنا بتقييم ثلاثة من نماذج الانحدار: حيث ظهرت أفضل النتائج لمجموعة الاختبار في نموذج XGBoost (MAE = 2.98 يوما، RMSE = 12.20 يوما، $R^2$ = 0.936، MAPE % = 14.43)، في حين كان أداء الغابة العشوائية أفضل من خط الأساس لنموذج الانحدار الخطي (MAE = 2.77، RMSE = 14.59، $R^2$ = 0.908، MAPE = 13.43% مقابل MAE = 18.48، RMSE = 24.57، $R^2$ = 0.740، MAPE = 51.50%). لتفسير هذه التنبؤات إلى معلومات عملية حول "وقت الاستغلال" المقدر إلى "مؤشر أولوية مركب" يجمع بين سرعة الاستغلال المتوقعة ودرجة الاحتمالية، ثم استخدم نظام تقييم إمكانات الاستغلال (EPSS) لتصنيف الثغرات الأمنية إلى المستويات التالية: حرجة وعالية ومتوسطة ومنخفضة. حسن هذا النهج قدرتنا على تحديد الثغرات الأمنية عالية الخطورة مبكرا من خلال دمج البيانات الزمنية، مقارنة بالاعتماد فقط على المعايير الثابتة، وتبين النتائج أن دمج البعد الزمني يعزز موثوقيتها وقابلية تطبيقها على نطاق اوسع.

**الكلمات الدالة:** الأمن السيبراني، تحديد أولويات الثغرات الأمنية، وقت الاستغلال (TTE)، نظام تسجيل نقاط التنبؤ بالاستغلال (EPSS)، نماذج الانحدار.

## 1. INTRODUCTION

Cybersecurity teams are getting with vulnerabilities discovered increasing day by day, it is becoming quite a task to take right and decisions regarding which vulnerabilities require immediate fixes.

In shadow of this significant accumulation traditional solutions that rely solely on static metrics such as the CVSS score are no longer sufficient to keep pace with the changing nature of threats.

Numerous studies have shown that traditional metrics are limited in their ability to predict the actual exploitation of vulnerabilities in operational environments [1], This deficiency dictates an urgent necessity for superior danger assessment approaches. While methods EPSS and CISA KEV have correctly mitigated some flaws, providing indicators of the possibility of exploitation or confirming its occurrence, they have shown a clear gap in their treatment of time as an outside element rather than a central element of the evaluation process [2]. It is not enough to rely exclusively on the severity of vulnerabilities as, for instance in traditional systems like CVSS, to address this problem emphasis was put on data and proposed a model (EPSS) based on community expertise and industry data for estimating the likelihood of exploitation [3]. That study is superior to other similar studies in a comparable manner, A novel assessment framework allowing a more realistic prioritization for remediation based on three indicators: exploitation chain risk, exploit code availability, likelihood of use was introduced, recognizing that CVSS does not adequately reflect exploitation risks in OT/ICS [4]. Using mid quantile regression and a novel accuracy metric (AGR) was presented in a statistical framework to incorporate uncertainty in vulnerability assessments. Proposed approach offers an assessment that is more robust and reliable even if some data is missing [5].

The regular CVSS scores do not limit the capabilities of basic CVSS but are not sufficient to represent the time-dependent properties of real-world scenarios. The authors relied on temporal metrics to Eq, to represent time-dependent properties. A Bayesian network may be used to calculate overall security status at a certain time

t. The model considers code availability and data trustworthiness to generate dynamic assessment of risk [2]. Also, a predictive system was created based on the graph network of vulnerabilities and vendors using RGCN, the features have been made richer by adding inputs such as vulnerability descriptions and exploit code availability. The results of this method in detecting high-risk vulnerabilities were more accurate than that of EPSS [6]. Still there is no consideration of the expected time to exploitation (TTE) in the assessment frameworks despite their contribution. Cybersecurity teams are struggling to prioritize vulnerabilities because of the continued expansion of threats. The core issue is that traditional scoring systems like CVSS, rely on static measures that fail to capture actual exploitation, while EPSS and CISA KEV, which indicate the likelihood of exploitation, fundamentally lack the integration of time as a key element in the assessment process. Therefore, this study introduces the development of a new model that integrates the "expected time to exploitation" mechanism as a key element for prioritization, which enhances the effectiveness of security teams' response in work environments.This study aims to design an integrated regression model that integrates multiple datasets to determine the expected time between vulnerability discovery and actual exploitation and incorporate expected time to exploitation in vulnerability prioritization to improve security teams' response to accelerating threats.Based on the above, the current study proposes a mechanism in which the expected time to exploitation is considered an important factor in prioritizing vulnerabilities. Achieving this requires developing a model based on learning from real-world data extracted from several sources, including severity scores. This model it will be work to estimate the expected time lag between the announcement of a vulnerability and its actual exploitation, which can later be used as a complement or alternative element in decision-making mechanisms. Thus, the study seeks to fill a significant gap in risk assessment tools by shifting from static assessment mechanisms based on general characteristics to dynamic ranking models based on a predictive time dimension.

This approach better reflects the reality of accelerating threats, helping cybersecurity teams make better decisions. [7]

## 2. MATERIALS AND METHODS

The practical part of this study focused on developing and evaluating regression models to predict the Time-To-Exploitation (TTE) for security vulnerabilities. This methodology section contains sufficient detail to reproduce the reported data and follows a rigorous modeling pipeline, leveraging time-series validation techniques. We used a dataset containing 93,110 vulnerabilities, collected from 2021 to 2024. The Python programming language was used for implementation, employing key libraries such as Pandas, NumPy, scikit-learn, and XGBoost.

### *2.1 Data Collection Method*

The approach was designed to ensure realistic predictive validity, treating the TTE task as a future prediction problem.

#### *Data Source*

To develop the model for this study, we relied on programmatically merging data from three separate data sets, which were combined using the CVE identifier key as the common key to obtain data from the National Vulnerability Database (NVD) that provided the metadata, CVSS scores, and Exploit Prediction Scoring System probability scores. [8] We also used actual exploitation data from CISA KEV and ExploitDB to determine the actual exploit dates. [9][10]

#### *Temporal Split*

A precise time-series cross-validation approach was implemented to assess the model's generalization capability on future, unseen data:

**Training Set:** Vulnerabilities covering the period 2021 to 2023.

**Test Set:** Vulnerabilities published in 2024.

***Target Variable Transformation***

To account for the skewed distribution of the TTE target variable and normalize its variance, a logarithmic transformation $\boldsymbol{y' = \log(1 + y)}$ was applied before training using scikit-learn's TransformedTargetRegressor. Predictions were reverted to the original time unit (days) using the exponential function ($\boldsymbol{expm(y)}$) .

### *2.2 Data Preprocessing and TTE Label Calculation*

A standardized data cleaning and feature engineering pipeline was followed to prepare the data for the regression algorithms.

***TTE Label Generation***

The target variable (TTE) was calculated as the duration in days between the vulnerability's publication date and its first observed exploitation date containing 87,576 labeled vulnerabilities. Two distinct TTE labels were calculated:

**Precise TTE:** To obtain actual exploit dates we used vulnerability data from CISA KEV and ExploitDB sources. [9][10]

**Extended TTE:** The final TTE label used for training, which combines actual data with synthetic estimates for unexploited vulnerabilities. The synthetic estimates were generated using Isotonic Regression to model the monotonic relationship between the EPSS score and observed exploitation dates, thereby filling missing labels and providing a robust training set.

These labels should therefore be interpreted as surrogate estimates rather than exact measurements of real-world exploitation timing.

***Feature Engineering***

The data was transformed into features suitable for machine learning, including:

**Quantitative Attributes:** CVSS Base Score, EPSS Score, and derived metrics like Impact/Exploitability Scores.

**CVSS Vector Attributes:** Textual components of the CVSS vector (e.g., Attack Vector (AV), Attack Complexity (AC), Privileges Required (PR), User Interaction (UI), CIA Triad, Scope) were converted into numerical representations.

**Temporal Attributes:** Publication year and publication month.

**Missingness Indicators**: Binary features were created to flag instances where original data values (e.g., CVSS Base Score or EPSS Score) were missing and subsequently imputed.

### *2.3. Statistical Models and Implementation*

Three main statistical and machine learning models were selected for TTE estimation and performance comparison:

**Linear Regression (LR):** Used as a simple baseline model.

**Random Forest Regressor (RFR):** Implemented as a robust ensemble model with a logarithmic transformation of the target variable using TransformedTargetRegressor.

**XGBoost (Gradient Boosted Trees):**
Employed as the primary advanced model due to its high predictive capability.
All models were implemented using the scikit-learn and XGBoost libraries.

Hyperparameter process respected the temporal nature of the data. Furthermore, the XGBoost algorithm was implemented with Early Stopping, a technique used to mitigate overfitting and find the optimal number of boosting rounds.

### *2.4. Performance Evaluation Metrics*

To rigorously assess the reliability and accuracy of the predictive models on the 2024 test set,

performance was evaluated using the following standard regression metrics:

**Mean Absolute Error (MAE):** Represents the average absolute difference between the predicted and actual TTE values (in days).
**Root Mean Squared Error (RMSE):** Measures the standard deviation of the residuals, penalizing large errors more heavily than MAE.
**Coefficient of Determination (Score):** Indicates the proportion of the variance in the TTE that is predictable from the independent variables.
**Mean Absolute Percentage Error (MAPE):** Quantifies the prediction accuracy as a percentage of the actual TTE value.

Model performance was analyzed not only across the entire test set but also broken down by year and stratified by CVSS severity levels to ensure comprehensive reliability assessment.

### *2.5. Robustness and Validation Framework*

We implemented a framework to enhance the robustness and reliability of the proposed model. This framework included the following procedures:

**Basic Training on Real Labels:** Training the model on accurate labels representing 1.05% of the data to assess the impact of artificial labels on the accuracy of the results.
**External Validation:** Evaluating the model's performance using data from a separate year (2020) to ensure that overfitting did not occur.
**Feature Removal:** Training the model without EPSS features to measure its independent predictive capability.

## 3. THEORY AND CALCULATION

### *3.1 Theoretical Framework*

The theoretical foundation of this study rests on the premise that Time-To-Exploitation (TTE) is a stochastic process influenced by multiple factors. This process is affected by both the inherent characteristics of the vulnerability (such as CVSS scores) and external, dynamic intelligence indicators (such as KEV and ExploitDB data).

**Deterministic vs. Probabilistic Elements:** Traditional systems like CVSS provide a static risk assessment, but fail to capture the assessment, but fail to capture the dynamic nature of exploitation. In contrast, EPSS offers a probabilistic perspective by estimating the likelihood of vulnerability exploitation within a specific timeframe.
**Hazard Function Model:** The integration of deterministic elements (CVSS, KEV, ExploitDB) and probabilistic elements (EPSS) forms the theoretical basis for building predictive models.

The hypothesis is that the exploitation event can be represented as a hazard function, where the probability of exploitation decreases gradually over time unless new indicators emerge (such as the release of proof-of-concept code).This theoretical framework aims to extend previous research by incorporating time-series machine learning models (such as Random Forest Regressor, XGBoost, and linear regression as a baseline) to better approximate the underlying statistical distribution of time-to-exploitation.

### *3.2 Calculation Methodology*

This computational methodology transformed this theoretical framework into practical steps for training and evaluating models:

**Defining the Time-to-Exploit (TTE) Label**
The time-to-exploit for each vulnerability is defined mathematically as follows*:*

$$\mathrm{TTE} = \boldsymbol{t}_{\mathrm{exploit}} - \boldsymbol{t}_{\mathrm{publish}} \quad (1)$$

where, $\boldsymbol{t}_{\mathrm{publish}}$ represents the vulnerability publication date, and $\boldsymbol{t}_{\mathrm{exploit}}$ represents the date of the first recorded exploit (from KEV, ExploitDB, or from the AI-derived estimate based on EPSS using Isotonic Regression).
The precise and extended TTE labels were used to enhance the training process.

***Feature Transformation***

All vulnerability attributes were encoded to suit regression algorithms:

**CVSS encoding:** The textual CVSS metrics (such as AV, AC, PR, UI, Scope, CIA) were converted to numerical values.
**EPSS integration:** The EPSS probability score was used (after handling missing values).
**Temporal attributes:** The publication year and month were extracted as temporal features.

***Model Training and Early Stopping***

Three main models were used, with a target variable transformation applied to the Random-Forest and XGBoost models using the logarithmic function (log(1+y)) to reduce variance.

**Linear Regression:** as a baseline for comparison.
**Random Forest:** as a non-linear ensemble model.
**XGBoost (eXtreme Gradient Boosting):** as an advanced model, using early stopping to prevent overfitting by monitoring the model's performance on the validation set.

***Evaluation Metrics***

The performance of the models was evaluated using standard statistical regression metrics, where $\widehat{y_i}$ represents the predicted value and $y_i$ the actual value:

**Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}|$$

(2)

**Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}$$

(3)

**Coefficient of Determination ($R^2$):**

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

(4)

**Mean Absolute Percentage Error (MAPE):**

$$\text{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \widehat{y_i}}{y_i}\right| \quad (5)$$

**Prioritization Function**

To facilitate practical application, the predictions were converted into a composite priority score.This function was designed to combine two key factors: the expected time to exploitation (TTE) and the global exploitation probability score (EPSS), as follows:$P = \alpha \cdot \frac{1}{1+\frac{max(0, TTE_{pred})}{\tau}} + \beta \cdot \text{EPSS}$

(6)

where $\tau$ represents the time decay constant, weights were assigned to the TTE ($\alpha$) and EPSS $\beta$) components, as well as the decay constant, using a differential evolution algorithm on the validation set to increase the accuracy of the selection at Precision@10%. A higher weight was finally assigned to the time component $\alpha$, since it is the most direct factor influencing the speed of response.

## 4. RESULTS AND DISCUSSION

This section aims to analyze the performance implications achieved by time-to-exploitation (TTE) prediction models, and discuss the implications of using layer-wise implementations, especially the reliance on synthetic tags and composite control.

### *4.1 Model Performance Evaluation*

The performance evaluation of the three models

showed clear performance differences, highlighting the effectiveness of the dependent variable. Logarithmic transformation and the combination of features used.

**Table 1.** Comparison of the performance metrics of the three models on the test set.

| Model | $R^2$ | MAE (Day) | RMSE (Day) | MAPE (%) |
|---|---|---|---|---|
| Linear Regression | 0.740 | 18.48 | 24.57 | 51.50 |
| Random Forest | 0.908 | 2.77 | 14.59 | 13.43 |
| XGBoost | 0.936 | 2.98 | 12.20 | 14.43 |

The results of the evaluation are clearly consistent (see Table 1). The XGBoost model achieves the best overall performance, primarily recording the highest agreement ($R^2 \approx 0.936$) and the lowest Root Mean Squared Error (RMSE ≈ 12.20) days. Then comes the Random Forest model with a slight difference, where the Mean Absolute Error was recorded (MAE ≈ 2.77 days), The linear regression model showed lower performance across all metrics ($R^2 \approx 0.740$, MAE ≈ 18.48). The large gap between the low MAE and the high (RMSE ≈ 12.20 days for XGBoost vs. MAE ≈ 2.98 days) indicates that the models, while accurate in most predictions, falter when faced with a small number of outliers with large errors. In addition, XGBoost and Random Forest models recorded relatively low MAPE values (around 14%), which is expected given the sensitivity of the relative error to forecasting very small time values.

To confirm the statistical superiority of the XGBoost model over the linear (basic) regression model and the Random Forest model, paired t-tests and Wilcoxon signed-rank tests were performed on the per-sample absolute errors computed on the same 2024 test set. The results showed that ($p < 0.001$), confirming that the performance improvement achieved was statistically significant. The 95% confidence interval for the mean absolute error (MAE) of the XGBoost model was also calculated, ranging from 2.833 to 3.136 days, with a mean of 2.979 days, further reinforcing the high confidence in the accuracy and reliability of the model's predictions.

### *4.2 Methodological Implications of Using Synthetic Tags*

This study represents an important analysis of the available data sources. The EPSS-derived synthetic tags represented 98.95% of the labeled dataset, compared to the actual tags which accounted for only 1.05%.

**Table 2.** Distribution of tags (Labels) by source

| Tag Source | Percentage (%) |
|---|---|
| Artificial Tags (Isotonic Regression based on EPSS) | 98.95% |
| Realistic Tags (KEV/ExploitDB) | 1.05% |

This distribution (see Table 2) shows the strong reliance on extended labeling using EPSS as a mechanism for generating supplementary tags in the absence of confirmed exploitation data. This serves as a practical demonstration of the use of EPSS. As a methodological tool, it also imposes a methodological limitation that requires caution. Although the model reflects a high prediction quality compared to another probabilistic model (EPSS), it cannot be considered an exact representation of real-world exploitation.

### *4.3 Methodological Challenge and Tag Coverage*

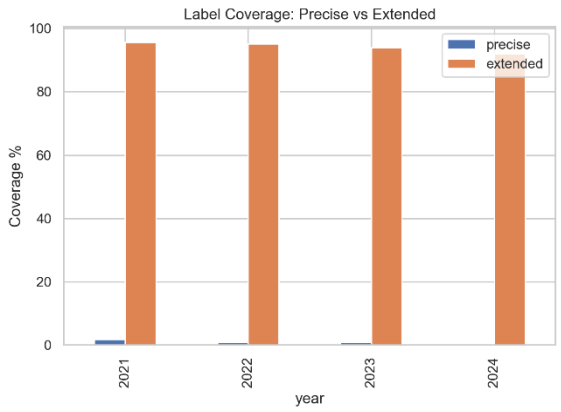


**Fig 1.** Comparison of percentage coverage of precise tags and extended tags by year.

Figure 1 shows that the high variance (98.95% vs. 1.05%) and reliance on expanded labeling (EPSS) is not a choice but a methodological necessity for conducting research in light of the scarcity of documented evidence of exploitation.

### *4.4 The Importance of Characteristics Prediction*

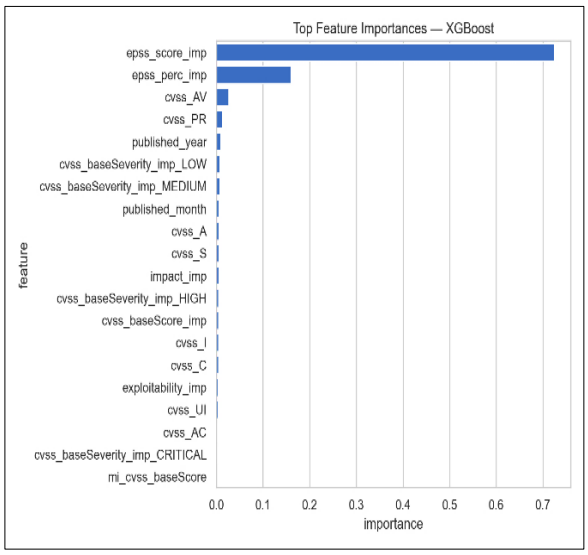


**Fig 2.** Importance of the predictive features of the XGBoost model (gain measure).

Figure 2 shows that the EPSS characteristics (score and percentage) are the strongest and most important predictors of Expected Time To Exploitation (TTE), clearly outperforming all CVSS characteristics.

### *4.5 Visual Prediction and Goodness of Fit*

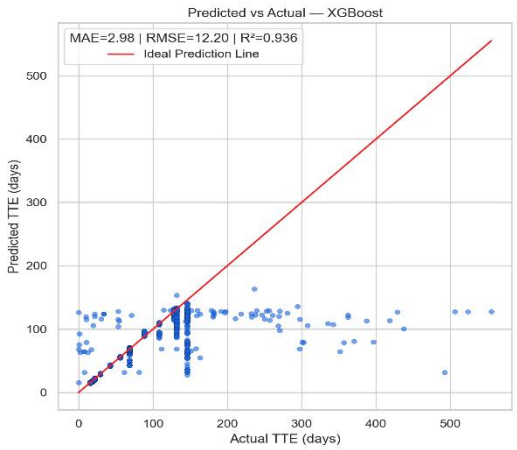


**Fig 3.** Scatter plot comparing predicted

Figure 3 shows that the points are clustered around the ideal prediction line (y = x), indicating a low discrepancy between the prediction and the actual value for this model. This explains the high value of the coefficient of determination ($R^2 \approx 0.936$) and the low value (MAE $\approx 2.98$) that the model recorded on the test set.

### *4.6 Performance of Linear Regression Model in Predicting TTE*

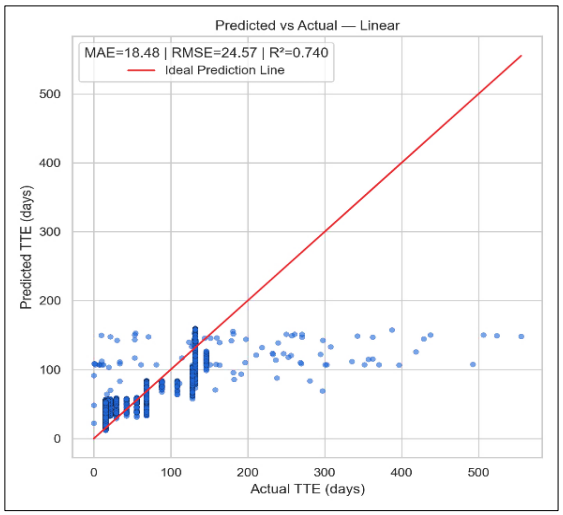


**Fig 4.** Predicted vs. Actual Value of a Linear Regression Model on the Test Set

Figure 4 shows that the linear regression model failed to predict the exploitation time and the blue dots representing the predictions are very scattered and far from the ideal line (y=x). This

large scattering explains why the model achieved 0.740 and this proves that a simple model cannot deal with this complexity of data.

### *4.7 Variance Analysis and Learning*

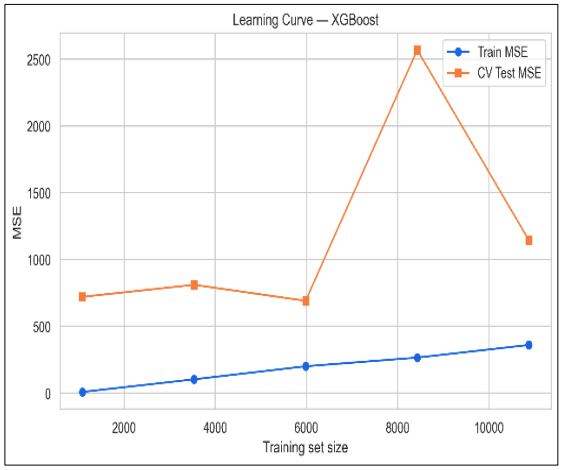


**Fig 5.** Learning curve for a XGBoost model: Mean square error (MSE) versus training set size.

Figure 5 shows that the significant gap between the training error and the cross-validation error (CV test) indicates that the model suffers from moderate variance and partially overfits the training data. The high value of the coefficient of determination shown in Figure 3 confirms the robustness of the input features.

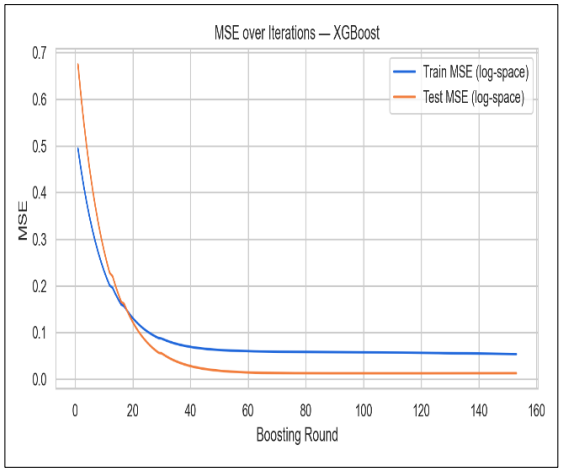


**Fig 6.** Mean Square Error (MSE) vs. Boosting Rounds XGBoost Model.

Figure 6 shows that the training error continues to decrease, which means that this model continues to save the training data, and the orange error, which represents the cross-validation error of its performance on new data, reached its best point and stabilized at around round 60. This is due to the use of the early stopping technique, which stopped training at this point to prevent excessive modification.

### *4.8 Absolute Error By CVSS Severity*

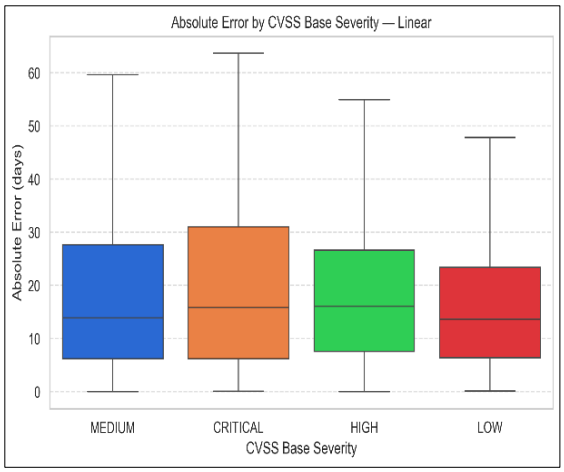


**Fig 7.** Absolute Error Based on CVSS Severity for Linear Regression Model

Figure 7 shows that the linear regression model covers large absolute errors. The mean error for each category ranges between approximately 15 and 20 days. This is considered poor accuracy for critical and high variations.

### *4.9 Absolute and Relative and Errors of Random Forest Model*

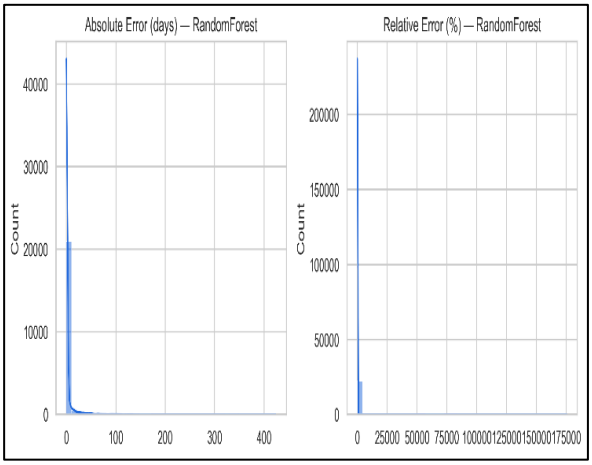


**Fig 8.** Distribution of Absolute and Relative and Errors of Random Forest Model

Figure 8 shows that the distribution of absolute and relative errors of the model and their clustering at zero, which is why the model recorded the lowest average absolute error (MAE) compared to other models.

### *4.10 Composite Priority Score vs. Time-To-Exploit*

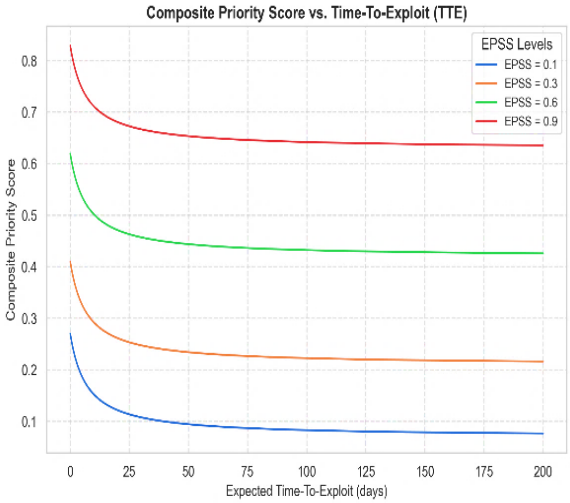


**Fig 9.** The inverse relationship between the composite priority score and the expected time to exploitation

Figure 9 shows a strong inverse relationship: the more quickly we expect a vulnerability to be exploited, the lower the TTE, the higher its priority. Then its importance decreases and stabilizes after about two months. The EPSS score determines the starting point for priority. Vulnerabilities that are more likely to be exploited (higher EPSS) start with a higher priority. The TTE element is what determines the speed at which the priority decreases.

### *4.11 Synthetic vs. Real Data Impact*

The comparison results showed that training the XGBoost model using limited real-world data (1.05% of the sample) resulted in a high mean error (MAE) exceeding 64 days, indicating the model's inability to generalize in the absence of sufficient data.

This limitation is mainly attributed to the scarcity of publicly confirmed exploitation timelines available in current vulnerability datasets. Therefore, extended labels were used to provide a sufficient sample size that allowed the model to learn the exploitation time patterns, significantly reducing the MAE to 2.98 days.

### *4.12 Feature Ablation Analysis*

To measure the substantial contribution of EPSS features to enhancing the model's predictive power, we conducted a feature ablation study by training the XGBoost model without these features. The results showed a significant performance gap, as illustrated in (see Table 3); the absence of EPSS features led to a sharp decline in accuracy, with the mean absolute error (MAE) increasing from 2.98 days to 37.83 days, and the coefficient of determination ($R^2$) dropping to 0.120. These results confirm that EPSS features are not merely supplementary inputs but rather the primary driver enabling the model to capture critical signals for exploit timing. However, the model's ability to maintain a certain predictive structure even in the absence of these features indicates its success in extracting additional patterns from other vulnerability characteristics, supporting our decision to combine these elements to produce a highly reliable composite priority score.

**Table 3.** Performance comparison with and without EPSS features

| Model Configuration | MAE (Days) | $R^2$ |
|---|---|---|
| Proposed XGBoost (With EPSS) | 2.98 | 0.936 |
| Ablated XGBoost (Without EPSS) | 37.83 | 0.120 |

### *4.13 External Validation Results*

To assess the model's generalizability beyond the training dataset, an independent test was conducted using 2020 data. The results showed a coefficient of determination ($R^2$) of 0.349, with a Mean Absolute Error (MAE) of 8.13 days. These results strongly indicate the model's ability to predict the timing of exploits in earlier

years not included in the training dataset, thus reducing the likelihood of overfitting and confirming the model's stability when applied to different time-series datasets.

### *4.14 Operational Simulation*

To demonstrate the practical value of the model in real-world environments, an operational simulation of the Top-K Remediation process was conducted. The results showed that using the proposed Composite Priority Score clearly outperformed traditional standards; the model achieved a recall rate (Recall@k) of 0.47, more than double the performance achieved using the CVSS standard alone (0.26). These results demonstrate the model's ability to help cybersecurity teams identify and remediate a larger proportion of high-risk vulnerabilities using the same available resources, thereby enhancing operational efficiency in threat management.

Despite the model's high accuracy, several methodological considerations must be addressed to ensure its continued effectiveness. First, there is the potential for "Labeling Bias" to transfer from the EPSS system we used to generate the extended labels. This could lead to predictions being influenced by the model's original probabilistic estimates rather than absolute field realities. Second, the rapidly evolving nature of cyber threats presents the challenge of "Model Drift".
The relationship between vulnerability characteristics and the timing of their exploitation can shift due to the evolution of attacker strategies. This necessitates continuous model updates and a mechanism for periodic retraining to ensure the model remains relevant and accurate over time.

## 5. CONCLUSIONS

This study concludes that moving beyond static metrics such as CVSS toward a dynamic predictive model that estimates the Expected Time To Exploitation (TTE) and integrates it into a "Composite Priority Score" combining TTE and EPSS, significantly improves vulnerability prioritization. The approach has proven effective in accelerating response through time-consistent regression models and the model's reliability was rigorously confirmed through statistical significance tests ($p < 0.001$) and external validation using independent datasets, which proved its generalizability and robustness against overfitting and quantitative assessments and improvements in operational parameters for handling vulnerabilities were also found achieving a Recall@k of 0.47 in the operational simulation($\boldsymbol{\alpha}$= 0.200, $\boldsymbol{\beta}$= 0.699,$\boldsymbol{\tau}$ = 7.0) where the decay constant $\boldsymbol{\tau}$ indicates that vulnerabilities expected to be exploited within the first seven days (one week) of their discovery should be given the highest priority, and is readily applicable for deployment in near-real-time monitoring systems. However, relying solely on the EPSS index is a limitation that may introduce time bias. Establishing a periodic retraining pipeline is also essential to mitigate potential model drift and ensure the model remains aligned with the evolving threat landscape.

Therefore, the study recommends collecting more real exploitation histories, conducting periodic statistical tests (such as significance tests and time recalibration), and external validation across different datasets, which would enhance the accuracy, reliability, and wider applicability of the results.

## 6. ACKNOWLEDGMENT

The authors would like to express their gratitude to everyone who provided assistance during the preparation of this study. We also appreciate the efforts of the open-source communities and data-sharing platforms that facilitated the data collection and analysis process.

## REFERENCES

[1] Jiang, Y.; Liu, T.; Wang, Y.; He, K. A Survey on Vulnerability Prioritization: Taxonomy, Metrics, and Challenges. arXiv preprint arXiv:2502.11070, Accessed on: Aug. 3, 2025. [Online].Available:

https://arxiv.org/abs/2502.11070

[2] Perone, S.; Guarino, S.; Faramondi, L.; Setola, R. Vulnerability Assessment Combining CVSS Temporal Metrics and Bayesian Networks. arXiv preprint, arXiv:2506.18715, 2025.https://arxiv.org/abs/2506.18715

[3] Jacobs, J.; Romanosky, S.; Suciu, O.; Edwards, B.;Sarabi, A. Enhancing Vulnerability Prioritization: Data-Driven Exploit Predictions with Community-Driven Insights. In 2023 Accessed on:Aug.5,2025.[Online]. Available: https://ieeexplore.ieee.org/abstract/document/10190703

[4] ArXiv Authors. Vulnerability Management Chaining:An Integrated Framework for Efficient Cybersecurity Risk Prioritization. arXiv preprint arXiv:2506.01220v3, 2025.

[5] Accessed on: Aug. 3, 2025. [Online]. Available: https://arxiv.org/abs/2506.01220

[6] Angelelli, M.; Arima, S.; Catalano, C.; Ciavolino, E. A robust statistical framework for cyber-vulnerability prioritisation under partial information in threat intelligence. Expert Syst. Appl., 2024. Accessed on: Aug. 7, 2025. [Online].Available:https://www.sciencedirect.com/science/article/pii/S0957417424014398

[7] Mahbub, M.; Khan, M. S. A.; Hamid, T.; Mia, M. S. A Novel Vulnerability Exploit Prediction System Using the Relational Vulnerability-Vendor Network.*Digital Threats: Research and Practice*,2024,6,no.2,pp.1–17**.** https://dl.acm.org/doi/full/10.1145/3724133

[8] Mell, P.; Spring, J. M. Likely Exploited Vulnerabilities, A Proposed Metric for Vulnerability Exploitation Probability. National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2025.
https://www.nist.gov/publications/likely-exploited-vulnerabilities-proposed-metric-vulnerability-exploitation-probability

[9] Manzoni, F. CVE, CISA KEV & EPSS Datasets. Kaggle, 2024 accessed on Aug. 9, 2025).https://www.kaggle.com/datasets/francescomanzoni/vulnerability-management-datasets

[10] Cybersecurity and Infrastructure Security Agency (CISA). Known Exploited Vulnerabilities Catalog. U.S. Department of Homeland Security, 2024 (accessed on Aug. 10, 2025). https://www.cisa.gov/known-exploited-vulnerabilities-cataloge

[11] Exploit-DB. files_exploits.csv. Git repository, GitLab, (accessed on Aug. 10, 2025).
https://gitlab.com/exploit-database/exploitdb/-/blob/main/files_exploits.csv