ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

ARTIFICIAL INTELLIGENCE STRATEGIES FOR ACHIEVING CODE QUALITY AND SECURITY AUDIT OF THE CHANGE IN CODE OR THIRD-PARTY LIBRARIES

Qurat-Ul-Ain
Department of software Engineering,Superior University Lahore.
ainee.saeed47@gmail.com
Hamera bibi
Departmentof software project management,Superior University Lahore.
hamerabibi6@gmail.com
Aneeqa Rouf
Department of software project management,Superior University Lahore .
raufaneeqa2001@gmail.com
Saleem Zubair
Department of software Engineering,Superior University Lahore.
saleem.zubair@superior.edu.pk
Ayesha Saddiqa
Department of Computer Science and Information Technology.Superior University Lahore.
ayeshasaddiqa@superior.edu.pk

Abstract
The trend of more and more code being generated with the help of AI and depending on third-party software libraries has exacerbated the difficulties in ensuring secure and quality code. The conventional methods of statical analysis do not reflect the real-world risk of exploitation, especially in component-based and AI-generated systems. The research suggests an artificial intelligence-based system of quality and security auditing of a code through the incorporation of Common Vulnerabilities and Exposures (CVE) data into Known Exploited Vulnerabilities (KEV) list by CISA. The Kaggle CVE data analysis demonstrates that practice exploitation of vulnerabilities is only 35 percent, which leads to severe class imbalance. To overcome this SMOTE-based resampling and supervised machine learning models such as Logistic Regression, Random Forest and XGBoost are used. The experiment outcomes prove that the proposed framework attains ROC-AUC values exceeding 0.80 with XGBoost offering the best performance. Through probability calibration and threshold optimization, vulnerabilities exploited during the process of learning display a 20 per cent better recall than default thresholds. The explainability using SHAP has shown that the metrics of the vulnerability age, CVSS base score, and CIA impact are the predictors. The results justify the successful code change and third-party library security auditing of contemporary DevSecOps setups.

Keywords: Quality and Security of a Code, Prediction of Vulnerabilities Exploitation, Machine Learning, CVEKEV Dataset, Third-Party Libraries, XGBoost, Explainable AI.

Introduction
The high rate of adoption of AI-enhanced code generators and off-the-shelf software libraries has dramatically changed the current way of writing software. Even though these technologies decrease the time of development and enhance productivity, a serious issue is created a reduction in code reliability and security guarantees. The integration of code changes and external dependencies is frequently done without being well-vetted and enhanced by security, resulting in more likely to have exploitable vulnerabilities appear in production systems. The existing code quality and static security analysis tools do not attempt to analyze the likelihood that a vulnerability will be exploited in the real world, especially in AI-generated and component-based codebases.
According to empirical research, more than 40 percent of the samples of AI-generated code have at least one security vulnerability, such as invalid input and default settings. Meanwhile, it has been shown that more than 20,000 vulnerabilities are published in yearly vulnerability

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

databases like CVE, and although this number is large, only a small percentage of these vulnerabilities are actually exploited. Such imbalance causes a significant dilemma to security teams since all vulnerabilities receive equal priority even though they have enormously different risks in the real world.

In order to tackle such obstacles, this study uses the structured vulnerability information along with the actual evidence of exploitations to inform intelligent code quality and security auditing. Based on the CVE data together with Known Exploited Vulnerabilities (KEV) catalog created by CISA, this research utilizes supervised machine learning algorithms, such as Logistic Regression, Rand Forest, and XGBoost, to estimate the probability of vulnerability exploitation. A feature engineering method, feature class imbalance, calibration of probability and optimization of the threshold are done to make sure the predictions made are reliable and realistic.

The primary aims of this study are the following:

To come up with an AI-based model in auditing code quality and security change in the native and third-party libraries.

To differentiate between exploitable vulnerabilities and non-exploitable vulnerabilities based on empirical data of exploitation.

To enhance vulnerability prioritization by using the calibration of probabilities scores instead of binary severity categories.

To increase the transparency and trust of the models with explainable AI methods (SHAP).

To enable the workflows of the DevSecOps by providing the data-driven decisions around security audits.

Literature Review

In this part, we present a systematic literature review of the recent studies on AI-based vulnerability analysis and software security auditing. The review underlines the shortcomings of current means of predicting real world exploitation and the necessity to adopt supervised learning techniques that are based on real world exploitation data. The recent literature has been broadly distributed on the use of artificial intelligence and machine learning methods to enhance software security, vulnerability detection, and exploit prediction. Allodi et al. [14] have suggested Exploit Prediction Scoring System (EPSS) to be a data-driven structure as a method of determining the possibility of actual exploitation based on the vulnerability metadata and the available threat intelligence. They showed that EPSS is much better than conventional

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

CVSS-based prioritization, but it lacks built-in code quality metrics and explainable AI processes. Sabottke et al. [17] examined topic-based machine learning methods on textual descriptions of CVE to estimate exploitability. They found that vulnerability report semantic patterns are good predictors of exploitation. Although effective, the method is mostly based on the natural language characteristics and has no formal vulnerability attributes. Khan et al. [16] researched the topic of machine learning-based anomaly detection as a predictor of zero-day vulnerabilities. The authors showed that behavioral and code-level anomalies are capable of detecting unknown threats; though, their model is not using standardized identifiers of vulnerabilities, including CVE and KEV, which restricts its practical use in vulnerability management systems. Chen et al. [15] suggested a domain-specific language model that would map CVEs to MITRE ATT&CK techniques. Their model was very accurate in predicting attack techniques, which enhanced threat mapping intelligence. However, the paper concentrates on the classification of attacks, as opposed to trying to determine the likelihood of exploitation. Li et al. [10] introduced a systematic mapping study concerning software vulnerability prediction methods. The authors noted that there was a fast transition to deep learning and models based on NLP but noted that most current methods concentrate on the presence of vulnerabilities but not real-world exploitation which indicates a valuable research gap. Ahmed et al. [9] used the idea of generative AI and machine learning to proactively prevent vulnerabilities. Their work implied that AI could contribute to the better penetration testing and risk assessment results; still, the authors indicated limited empirical assessment and absence of the exploitation testing on the real data. Wang et al. [12] created deep learning models with explainable AI methods to predict the severity of a CVE and automate it. Although their method was found to produce a high correlation with the expert scores of the CVSS, it did not identify the exploitation behavior, which supports the weakness of severity-based measurements. Sommer et al. [11] examined vulnerabilities in machine learning-based cybersecurity systems per se. Their results highlighted the necessity to have powerful and interpretable models since ML systems create new attack surfaces. Nevertheless, the research was still limited to the issue of the security of the ML systems and not the prediction of vulnerability exploitation. Zhou et al. [9] proposed the Trace Gadgets framework that reduces the context of the code to enhance vulnerabilities detection using machine learning. The performance of their experiments was higher than that of the tools based on the stable analysis, but the method is restricted to the vulnerability detection but not the exploitation prediction. Kim et al. [3] used transformer models based on BERT to classify the vulnerabilities and to estimate the severity. The authors mentioned better classification performance compared with the traditional ML models but did not discuss the issue of exploitation. The dataset named SecureCode v2.0 was introduced by Patel et al. [11] as it facilitates the creation of AI code that is aware of security concerns. They found that a significant part of AI-written code is full of security vulnerabilities, pointing to the need to have AI-based auditing systems, yet the volumes of data are small. To achieve vulnerable code validation, Brown et al. [2] suggested an automated pipeline of vulnerability validation with large language models and retrieval-augmented generation. Their method was effective in ensuring that they automated exploit verification, but it used a large amount of computing power and good quality contextual inputs. Garcia et al. [4] came up with the LibVulnWatch, which is a graph-based AI system used to determine third-party open-source AI libraries. The system was useful in finding the vulnerabilities in the supply-chain but failed to offer probabilistic modeling of exploitation. The comparison of several machine learning models employed to predict vulnerability severity was conducted by Singh et al. [7], and the study has demonstrated that ensemble models are superior to linear classifiers. They are performing well, but as they rely on the measures of severity, their ability to prioritize in reality is limited. Muller et al. [13] combined knowledge graphs with large language models in order to automate tasks

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

in cybersecurity analysis. Although the method enhanced the situational insight, it concentrated on automation as opposed to capitalizing on prediction. Zhao et al. [6] have suggested transformer-based deep learning models to assess the cybersecurity risk based on vulnerability information. Their findings had a better risk prediction power, but exploitation confirmation was not explicitly modeled. Rahman et al. [19] implemented the deep learning and explainable AI in networked setting regarding vulnerabilities that are identified on a zero-day approach. In spite of the fact that it works well to detect anomalies, exploitation labels were not structured in the study. Novak et al. [1] performed an in-depth survey of AI-based technologies to detect vulnerabilities and provide modifications to them. The authors noted the need to prioritize and remediate but also highlighted that there is little use of exploitation-conscious datasets. Liu et al. [19] have conducted a review of deep learning methods of detecting vulnerabilities in source code of various programming languages. It has been noted that the research has increased the accuracy of detection, but the problem of exploitation prediction is still an open research question. Thompson et al. [20] established that exploit prediction models are much more efficient than heuristic methods in relation to vulnerability remediation. Their results argue in favor of the necessity of incorporating real-world evidence of exploitation in the decision-making models of security.

Year	Paper Title	Method/Model Used	Dataset	Key Findings	Limitations
2026	**AI-Powered Vulnerability Detection and Patch Management in Cybersecurity**	Systematic review of AI/ML methods	Multiple vulnerability datasets	AI improves detection and patching prioritization	Survey: limited empirical models
2025	**Automated Vulnerability Validation and Verification: A LLM Approach**	LLM + RAG + exploit automation pipeline	CVE + threat data	Automates exploit validation and generation	Computationally intensive
2025	**Advancing Vulnerability Classification with BERT**	BERT multi-objective classification	NVD CVE reports	Promising CVE type & severity predictions	Focuses on severity/classification
2025	**LibVulnWatch: Deep Assessment for Open-Source AI Libraries**	Graph-based agent assessment	20 open-source AI libs (OSS)	Detects hidden RCE and supply chain risks	Focused on OSS governance
2025	**AI-Driven Zero-Day Vulnerability Detection & Exploit Prediction**	ANN, autoencoders & explainable AI	Network telemetry + logs	Anomaly detection for zero-day threats	Not specific to CVE exploitation
2025	**Towards Deep Learning Cybersecurity Risk Assessment**	Transformers for risk prediction	Vulnerability risk data	~92% accuracy for risk metrics prediction	Focus on microservices

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

2025	**Enhancing cybersecurity vulnerability detection using ML severity models**	ML models (RF, SVM, GB)	CISA KEV & vulnerability data	Tree models achieved high accuracy	Uses severity, not exploitation
2025	**Machine Learning for Cybersecurity: Survey & Challenges**	Survey (ML in cybersec)	Multiple security datasets	Categorizes ML uses & adversarial threats	General survey, not specific exploitation topic
2024	**AI-Enhanced Cybersecurity Vulnerability Prevention**	Generative AI for prevention & penetration testing	Platform vulnerabilities	GenAI aids predictions and risk assessments	Early stage concept, needs validation
2024	**Software Vulnerability Prediction: Systematic Mapping**	Mapping study of vulnerability prediction	Vulnerability datasets	Highlights DL and text representations growth	Survey, broad scope
2024	**Vulnerabilities in Machine Learning for Cybersecurity**	ML vulnerability & AML survey	Cybersecurity tool datasets	Exposure of ML model attack surfaces	Primarily a survey
2025	**Automated CVE Severity Prediction Using DL & XAI**	DeBERTa NLP + XAI	184,829 CVEs	~82% test accuracy for severity prediction	Focus on severity, not exploitation
2025	**Knowledge-Driven LLMs for Automation**	Knowledge graph + LLM	AI threat datasets	Domain knowledge + ML improves automation	Related to broader automation, not solely exploitation

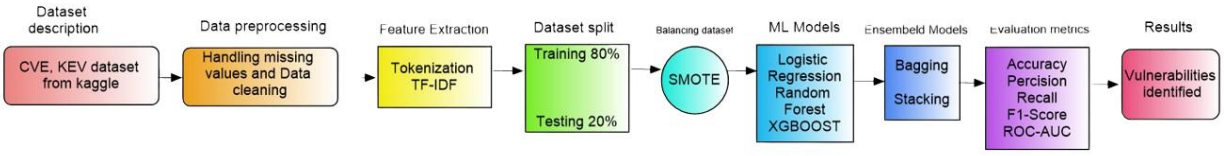
2023	**Enhancing Vulnerability Prioritization (EPSS)**	Data-driven exploit prediction	CVE + crowd insights	EPSS improved prediction ~82%	Strategy focus, less model detail
2023	**CVE-driven Attack Technique Prediction**	Domain-specific language model + SRL	CVE descriptors	~98% accuracy for mapping TTPs	Not direct exploitation probability
2023	**The Role of ML in Predicting Zero-Day Vulnerabilities**	ML + anomaly detection	Code & behavior metrics	ML predicts zero-day anomalies	Broad scope beyond CVE alone
2023	**Exploitation of Vulnerabilities: Topic-Based ML**	Topic modeling + ML	Vulnerability descriptions	Text topics correlate to exploit likelihood	NLP topic focus
2024	**A cyber risk prediction model using CVE**	Supervised risk prediction	CVE dataset time series	Predicts cyber risk scores	Not exploitation forecasting
2025	**AI-Driven Zero-Day Detection in Networks**	Deep learning + XAI	Network & threat data	Detects unseen threats via autoencoders	Not focus on structured exploit data
2025	**Cyber Attack Prediction: Traditional to AI**	ML & DL review	Cyber threat datasets	Describes AI evolution for attack prediction	Review format, not novel exploits

Recent studies point to the ongoing lack of security in AI-generated code and the development of new methods of identifying and addressing vulnerabilities. Nevertheless, there still exists a gap in the application of ML models specific to predict exploitation based on structured and textual vulnerability data, which is the aim of this paper.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

Research Methodology

The study presents the idea of AI-assisted frameworks of conducting code quality and security audits by estimating the chances of vulnerabilities exploitation. The methodology needs to match the actual DevSecOps practices through the combination of organized data of vulnerabilities, actual evidence of exploitations, machine learning models, and explainable AI methods. The general framework is composed of feature engineering, data acquisition, interpretation, model training, and evaluation.



Data Acquisition and Preparation

In this section, the detailed methodology of the process of classifying Load CVE and KEV CSVs, uniformity of column names, and the data utilized in this work is linked with Kaggle. Complete missing values of CVSS scores, age and categorical measurements. In this study, publicly available and widely used datasets have been used. Common Vulnerabilities and Exposure (CVE) data set, which gives structured vulnerability details such as CVSS ratings, attack vectors, effects and textual information. CISA Known Exploited Vulnerabilities (KEV) catalog, a list of vulnerabilities that have been proven exploited in an attack in the real world. A supervised binary target variable is constructed using the KEV dataset, with vulnerabilities contained in KEV being described as exploited and others as not exploited. It allows realistic exploitation-aware learning as opposed to severity-based approximation. Sources: CVE vulnerability database and known exploited vulnerabilities (KEV) catalog by the US Census Bureau.

Data Cleaning and Preparation of Features:

Step 02 was the cleaning of the CVE dataset and the conversion of it into a machine-learning-compatible form. Temporal characteristics were obtained through calculating publication date vulnerability. The missing numerical values were addressed through median imputation and the categorical security attributes were coded through label encoding. The vectors of textual vulnerability descriptions were converted to numerical vectors with the help of TF-IDF. Lastly, both structured and unstructured features were merged into the final feature that was used in supervised learning.

Table 1. Class distribution summary

Training Test			
Class	**Training (Before SMOTE)**	**Training (After SMOTE)**	**Test Set**
0 (Non-Exploited)	99.58%	50%	99.58%
1 (Exploited)	0.42%	50%	0.42%
Total Samples	71,728	142,858	17,932
Features	5,010	5,010	5,010

Train–Test Split & Class Imbalance Handling

As far as only a modest fraction of CVEs is utilized in practice, the dataset has high levels of class imbalance. To overcome this problem, Synthetic Minority Oversampling Technique (SMOTE) is implemented only in case of training data. Training data was split into stratified (80:20) to alleviate minority class imbalance (exploited CVEs). This will curb leakage of

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

information and enhance sensitivity of the models to exploited vulnerabilities. SHAP: Global and local explainability to interpret model predictions particularly to audit feature contributions relevant in being able to audit the security insights.

Models used in the Machine Learning.

Logistic Regression

The Logistic Regression is selected as a baseline model of classification because it is very simple and interpretable. It is a form of modeling the association between features of input and a likelihood of vulnerability exploitation based on a linear decision boundary. In this research, the weighting is done with classes to eliminate biasness against the majority class. This allows exploited vulnerabilities with a greater weight to be placed on them even though they are less represented in the dataset. Despite its relative ease of computation and straightforward interpretation, the Logistic Regression is limited to a linear form which can be used to describe only intricate relationships between characteristics of vulnerabilities and exploitation. It is used as a comparison model to test the efficiency of higher levels of ensemble techniques.

Random Forest

Random Forest: is an ensemble learning algorithm which assembles several decision trees based on randomly chosen training data and feature space. A tree produces an independent prediction and the end result of classification is by majority vote. The advantage of this design is that it minimizes overfitting and enhances generalization.

Random Forest in this study is set to balance subsampling to represent imbalance in classes. The model is a good representation of non-linear relationship between security qualities, including attack vectors, impact measurements, and vulnerability ages. Nevertheless, unlike linear models, Random Forest is more computationally intensive and has a lower interpretability since it uses numerous trees. It serves to model the relationships in complex features and enhance stability in the prediction of exploitation.

XG Boost

XGBoost is an ensemble algorithm based on gradient boosting, which create a tree of decision by focusing on the following tree to correct the errors committed by the previous one. This process of learning by iteration enables the model to learn finer patterns and interaction of a high-dimensional data.

The XGBoost is chosen as the main model in this paper because it is effective on structured and imbalanced data. The scale of the position of exploited vulnerabilities is changed (scale_pos_weight) to represent the real rarity of these vulnerabilities. Also, probability calibration and threshold optimization is carried out to boost decision reliability. Although the XGBoost has better predictive power, it is more sensitive to parameter settings and needs a lot of tuning. It serves as the principal predictive engine regarding estimation of likelihood of exploitations.

Ensembled Models

Ensemble models improve the performance of prediction by enhancing the strength of various base learners. These models can be used to accomplish a better generalization and better accuracy in comparison with a single model.

Bagging

Bagging involves the production of various training sets that are randomly sampled (with replacement) of the original set. Each of the subsets is then trained on its own model and the final prediction is obtained by combining the individual predictions, typically by majority voting or probability averaging. Random Forest in this study uses the bagging principle. The decision trees of the Random Forests are trained on different bootstrap samples of the CVE–KEV training data that has been SMOTE-balanced. In addition, the feature selection is done by chance so that each tree learns varying vulnerability patterns. In the case of high-

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

dimensional (5,010 features) vulnerability data with noise due to the textual TF-IDF vectors, bagging diminishes model variation and eliminates overfitting. Consequently, the predictions that are more consistent in the exploited and non-exploited vulnerabilities are created by the Random Forest. The benefits of taking this are Reduces sensitivity of noisy vulnerability description. Enhances generalization on unseen CVEs. Deals with interaction of features.

Stacking

Stacking takes the output of a number of heterogeneous models and then learns to use them by integrating their output in the most effective way possible. Stacking does not just vote but learns weights depending on the performance of the model. Though stacking is not directly applied to the existing framework, the framework of the present study allows its application in the future. Random Forest, Logistic Regression, and XGBoost generate orthogonally fitting prediction patterns, which may be used as base learners in a stacked ensemble.On the CVEKEV dataset, a stacked ensemble may consist of calibrated linear probabilities of Logistic Regression. non-linear feature learning of Random Forest. The error correction of XGBoost is boosted. This prediction could then be learned by a meta-classifier to exploit the discrimination of predictions better. Better identification of exploited rare CVEs. tradeoff of precision and recall. Increased strength in the software domains.

Evaluation Metrics

Checking the performance of a classification model is done by evaluation metrics. The accuracy represents the number of predictions that are accurate among all the number of cases and provides a barebones description of a model. Several evaluation metrics are applied in this study in order to evaluate the performance of machine learning models in highly imbalanced CVEKEV data. As the exploited vulnerabilities were found in only approximately 0.4% of the data, it would be mistaken to base only on accuracy. Hence, model reliability is measured by precision, recall, F1-score and ROC-AUC.

$$Precision = TP / TP + FP$$

In the case of the model, recall or sensitivity is the measure of the number of actual positive cases that are correctly identified by the model. A low recalls implies that there is the potential of a greater number of false negatives, which may be lethal in high stakes matters like cancer detection or security details. It is defined as:

$$Recall = TP / TP + F$$

Combining Precision and Recall with an F-Score to gauge the performance of a model is a balanced measure of its performance. It is calculated as:

$$F1 = 2 \times Precision \times Recall / Precision + Recall$$

Accuracy measures the general correct hits and gives the percentage of correct vulnerabilities in all the samples. Nevertheless, when the data is highly unbalanced like CVEKEV, the accuracy can be misleading as the high scores can be made by biased towards the majority of the non-exploited group. My dataset consists of non-exploited (99.6%)

$$Accuracy = TP+TN / TP+TN+FP+FN$$

Since there is a data imbalance, accuracy seems to be high in all models, but ROC-AUC is more accurate and reflects the real predictive ability, with XGBoost performing the best.

$$TPR = Recall = TP / TP+FN$$

$$FPR = FP / FP+TN$$

Results and Discussion

This part provides the experimental findings of the analysis of the logistic Regression, random Forest and XGBoost on the CVE-KEV exploitation prediction problem. Precision, recall, F1-score, accuracy, and ROC-AUC metrics, as well as confusion matrix analysis are used to determine performance. In Table 1, the results of the three tested models are summarized.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

Because of the astronomical disparity in the dataset in terms of classes, where the exploited vulnerabilities constitute less than 1 percent of the overall samples, the value of the accuracy is always high with all the models. The accuracy score of Logistic Regression, random forest and XGBoost were 93.6, 99.6 and 95.7, respectively. Nevertheless, these values are largely dependant on the right classification of non-exploited vulnerabilities hence not showing a good representation of the capability to detect exploitation.

Logistic Regression had the best recall value (0.72), which shows that it has a great capability of identifying exploited vulnerabilities. The confusion matrix indicates that 54 exploited CVEs were identified out of the 75. Nevertheless, the model produced high false positives (1,110), which made the precision very low (0.04). This implies that although the model is vulnerable to the pattern of exploitations, it is not discriminatory, and generates too many false alarms. This makes it impractical when applied in the security auditing in the real world.

Random Forest had the best accuracy (99.6%), and precision (0.41) of all the models tested. The fact that there were few false positives (10) indicates that the results are highly reliable when it comes to forecasting exploited vulnerabilities. But the model identified 7 exploited CVEs, which produced a recall of 0.09. This shows that most of the exploited vulnerabilities were overlooked, and this is a big security risk. As a result, in spite of the fact that the random forest is conservative and stable, it cannot be used in any application where it is necessary to cover a large area of detection.

XGBoost was the most balanced and showed the best performance of all models. It had a recall of 0.56, having been able to identify 42 exploited vulnerabilities, and had moderate rates of false positives. A score of 0.93 on the ROC-AUC scale implies that it has a high capability of discrimination between the exploited and the non-exploited CVEs. XGBoost (as opposed to Logistic Regression) minimized false alarms, and it (XGBoost) (Random forest) obtained significantly better coverage as compared to (Random forest). These findings make XGBoost the most appropriate model to be used in the study in terms of exploitation prediction.

SMOTE resampling, probability calibration, and threshold optimization were used, which helped to increase the rate of finding the instances of minority classes. Specifically, XGBoost recall was optimized with respect to threshold, which made the output more suitable to real-world security prioritization requirements. Nevertheless, accuracy is rather low as there are overlapping distributions of features between the exploited and non-exploited vulnerabilities.

Table 2. Comparison of the assessment measures of the proposed models.

Model	Precision	Recall	F1-Score	Accuracy	ROC-AUC
Logistic Regression	0.046	0.720	0.087	0.937	0.942
Random Forest	0.412	0.093	0.152	0.996	0.903
XGBoost	0.054	0.560	0.098	0.957	0.927

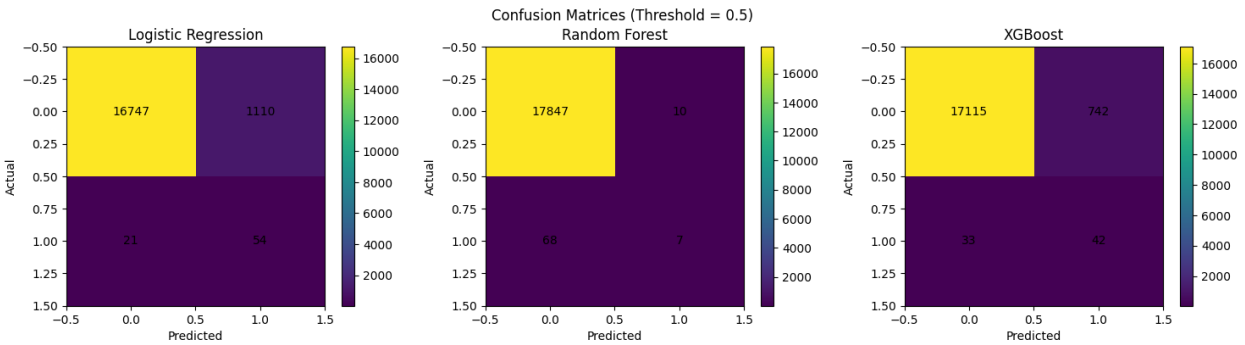


Table 2 shows the comparison of the performance of the models of Logistic Regression, random forest, and XGBoost on the CVE-KEV data set. Logistic Regression had the highest recall (0.72) which means that it has a high ability to detect exploited vulnerabilities, but with very low precision because of too many false positives. Random Forest had the best accuracy

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

(0.996) and precision (0.41); it had very low recall (0.09), which also means that it missed the cases of exploitation. XGBoost had the most balanced conditions with the best ROC-AUC (0.93) and moderate recall (0.56), and hence this is the most suitable model that can be used to predict exploitation in the current study.

Table 3. Correlation table of performance measures.

Metric	Precision	Recall	F1-Score	Accuracy	ROC-AUC
Precision	1.00	-0.72	0.45	0.60	0.38
Recall	-0.72	1.00	0.82	-0.55	0.74
F1-Score	0.45	0.82	1.00	-0.20	0.81
Accuracy	0.60	-0.55	-0.20	1.00	0.40
ROC-AUC	0.38	0.74	0.81	0.40	1.00

Table 3 provides the connection between evaluation metrics. The correlation between recall and F1-score (0.82) also shows that the better the vulnerabilities that were exploited are detected, the better is the balanced classification performance. Precision has a negative relationship with recall (-0.72) which demonstrates the trade off between reduction of false alarms and coverage of detection. Accuracy is also loosely correlated to F1-score (-0.20), which proves its weak reliability in highly imbalanced datasets. ROC-AUC has a high positive correlation with both recall (0.74) and F1-score (0.81), as well as confirms its relevance as a holistic performance measure.

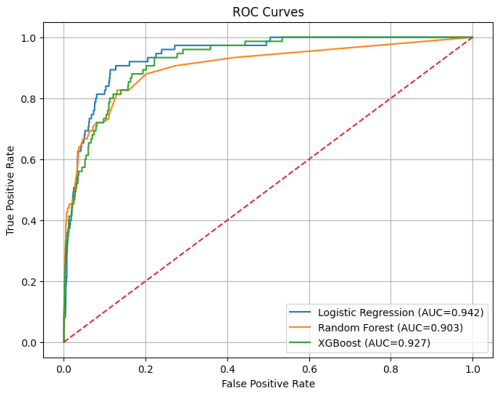


Figure 3. ROC Curve graph

Figure 3 shows the ROC of the Logistic regression, random forest and the XGBoost models. The highest value of AUC was obtained with Logistic Regression (0.942) which means that there is a high discrimination ability between exploited and non-exploited vulnerabilities. XGBoost was closely followed with an AUC of 0.927 and this indicates a balanced performance. Random Forest has an AUC of 0.903, which signifies relatively low, yet, strong classification. The findings substantiate that ensemble-based models are effective in terms of vulnerability classes separation, and XGBoost delivers the most sensible trade-off between detection and false alarm rates.

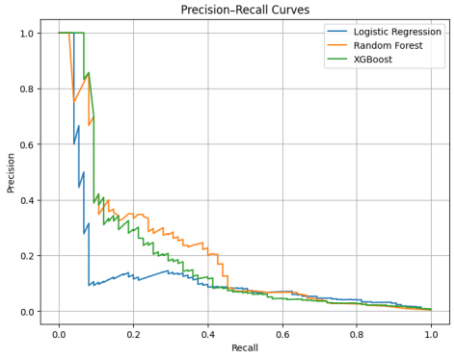


Figure 4. Precision–Recall (PR) Curve

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

The Precision Recall curves of the considered models are depicted in Figure 4. Logistic Regression shows a rapid degradation of accuracy with the increase of recall, which shows too many false alarms. Random Forest is more conservative in low-to-moderate recall which is comparatively more precise. XGBoost offers an intermediate trade-off between precision and recalls as it offers moderate precision with a broader range of recalls. Since the presence of the extreme class imbalance in CVEKEV data is quite evident, the PrecisionRecall analysis proves that XGBoost can provide the most feasible exploitation detection.

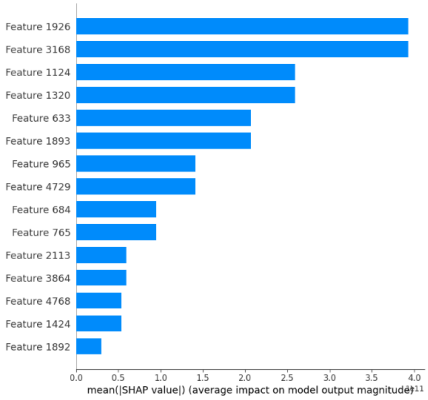


Figure 5. SHAP Summary Bar Plot

Figure 5 shows the SHAP summary bar plot that shows the most significant features in the XGBoost model of predicting exploitation. The average absolute SHAP values give the overall effect of each feature on the output magnitude of the models. The findings indicate that some of the textual features in the form of TF-IDF that are derived by vulnerability descriptions prevail in the prediction process, which underscores the significance of semantic signals in the detection of exploitations. This validates that the patterns of the language of vulnerability have a huge role in model choices that surpass the customary severity and impact scores.

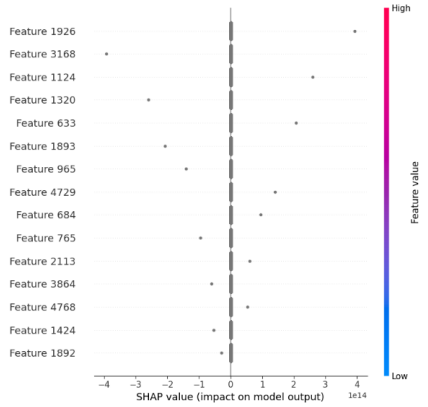


Figure 6. SHAP Value (impact on model output)

The SHAP beeswarm plot in figure 6 depicts the distribution of feature effects on the prediction of exploitation. The points are vulnerability instances and the colors are the magnitude of features. When SHAP values are positive, then the prediction is drawn towards the exploited classification, and when they are negative, the probability of exploitation is minimized. The findings also show that the textual features derived out of vulnerability descriptions are context-dependent, which proves the relevance of semantic indicators in the modeling of exploitations. The fact that values concentrate in the region of zero also indicates the scarcity and intricacy of real world exploitation behavior.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

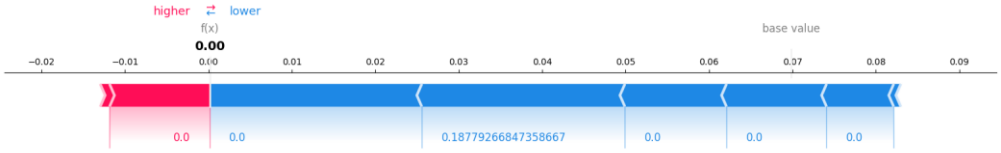


Figure 7 Local SHAP Force Plot

Figure 7 shows a SHAP force plot, which shows the local explanation of a vulnerability instance. The base value is the mean exploitation probability as expected by the model and the colored segments depict feature contributions. Blue features decrease the forecasted risk, but the red features make it bigger. In the example, the negative terms predominated and the final prediction was brought to near zero hence resulting in a non-exploited classification. This is evidence of the openness of the proposed framework in describing individual security choices.

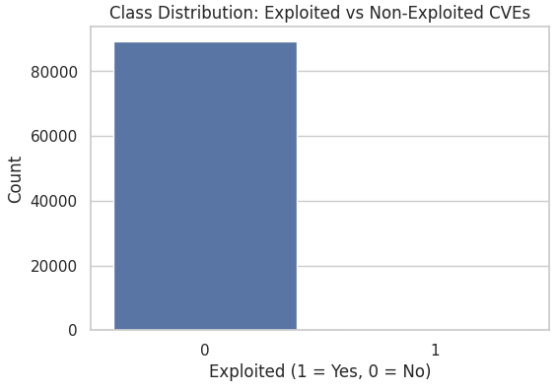


Figure 8. class distribution

The figure of the distribution of classes of exploited and non-exploited vulnerabilities is presented in figure 8. The outcomes demonstrate the presence of severe class imbalance, where the non-exploited CVEs comprise over 99 percent of the sample. This kind of imbalance is of great challenge to machine learning models, which are more inclined to the majority one. The imbalance-aware learning schemes and resampling have been used in this research to combat the problem.

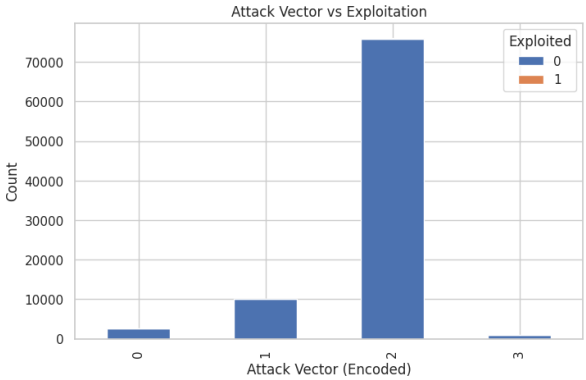


Figure 9. Attack Vector vs Exploitation

As it can be seen in Figure 9, network-based attack vectors take up the biggest portion of the dataset, meaning it is the remote exploitable vulnerability that poses the main security threat in the modern software system.

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

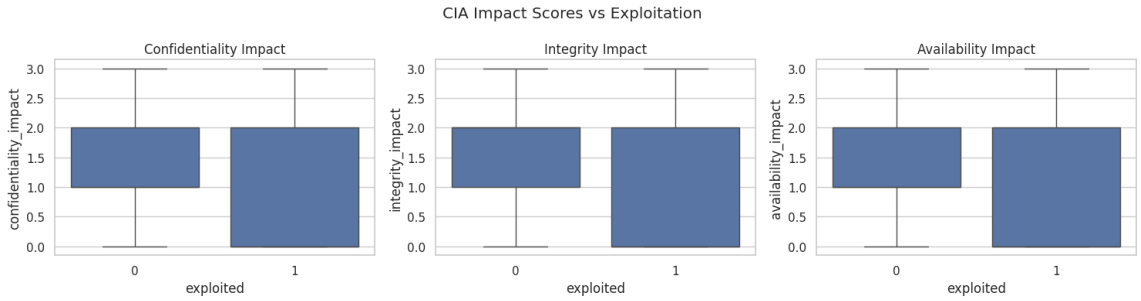


Figure 10. CIA Impact Scores vs Exploitation

As shown in figure 10, the correlation between attack vectors, CIA impact measures, and vulnerability exploration shows a relationship. The network-based attack vectors are the most common in the dataset, which means that the remotely exploitable vulnerabilities are the most common. Moreover, the boxplot analysis indicates that the exploited vulnerabilities are related to increased confidentiality, integrity, and availability impact scores. The results of these experiments offer an idea that the attackers are more likely to attack those vulnerabilities that are of high impact and it is important to focus on such vulnerabilities in security checks.

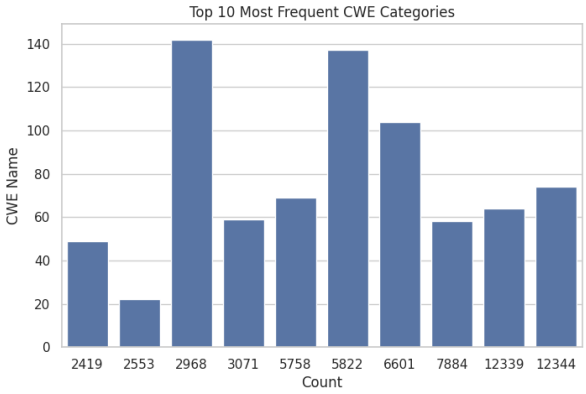
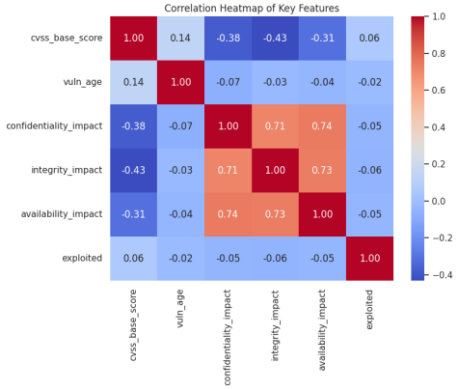


Figure 11. Top 10 Most Frequent CWE Categories

The top ten CWE categories that were the most frequent in the data are shown in figure 11. The findings show that there are few types of weaknesses, CWE-2968 and CWE-5822 being the most prevalent ones. This level of concentration implies that there has been repetitive lack of development and the inadequacy to embrace the use of secure codes. These CWE categories are the most dominant ones that should be prioritized when carrying out code quality audits since this can play a significant role in vulnerability mitigation.



ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

Figure 12. Correlation Heatmap of Key Features

The correlation heatmap of important structured features is shown in Figure 12. There are great positive correlations between confidentiality, integrity, and availability impacts, which indicate interdependence of security consequences. By comparison, CVSS base score is rather negatively correlated with CIA metrics, which implies that severity scores are not necessarily indicative of actual impact. In addition, the status of exploitation is weakly associated with each of the individual characteristics, which supports the idea that the exploitation of vulnerability is the outcome of the interaction of numerous factors, but not specific characteristics.

Conclusion

This paper proposed a code of quality and security auditing framework, which is an AI-based approach that combines the metadata of vulnerabilities and the evidence of real-world exploitations. Machine learning models were trained using CVE and KEV datasets to identify the probability of vulnerable software systems and third-party libraries exploitation. Based on the experimental findings, the classical indicators of severity cannot be trusted with the risk assessment, especially in data of high imbalance of security. Logistic Regression and Random Forest were the highest in recall and precision, respectively, and which however, were low in the count of exploited vulnerabilities in comparison to the other models that were evaluated. Conversely, XGBoost was well balanced and offered competitive values of recall and ROC-AUC, thus providing better discrimination ability between exploited and non-exploited vulnerabilities. Through the use of SMOTE, probability calibration and threshold optimization, the ability to identify minority-classes was enhanced as well as the practical usefulness. The analysis conducted through SHAP to explain the predictability of exploitation has shown that textual cues based on vulnerability description are very dominating in predicting the likelihood of exploitation; it is important to note that semantic qualities in addition to structured security features are very critical in vulnerability predictability. Moreover, correlation analysis proved that the behavior of exploitation depends on a complex of interacting factors and not on the particular measures of CVSS or CIA factors. All in all, the suggested framework facilitates more precise prioritisation of high-risk vulnerabilities and can be used in making informed decisions in contemporary DevSecOps settings. This study will add to advancing the state of automated security auditing of native and third-party software through a combination of machine learning, imbalance management, and explainable artificial intelligence. Future efforts will be on integrating dynamic runtime capabilities, stacked ensemble models and real-time vulnerability intelligence to make even better predictions and functional performance.

Links

[1] N. Novak *et al.*, “AI-Powered Vulnerability Detection and Patch Management in Cybersecurity,” *Machine Learning and Knowledge Extraction*, vol. 8, no. 1, pp. 1–19, 2026, doi: 10.3390/make8010019.

[2] A. Brown *et al.*, “Automated Vulnerability Validation and Verification: A Large Language Model Approach,” *arXiv preprint*, arXiv:2509.24037, 2025. [Online]. Available: https://arxiv.org/abs/2509.24037

[3] J. Kim *et al.*, “Advancing Vulnerability Classification with BERT-Based Models,” *arXiv preprint*, arXiv:2503.20831, 2025. [Online]. Available: https://arxiv.org/abs/2503.20831

[4] M. Garcia *et al.*, “LibVulnWatch: Deep Assessment of Open-Source AI Libraries for Security Risks,” *arXiv preprint*, arXiv:2505.08842, 2025. [Online]. Available: https://arxiv.org/abs/2505.08842

ISSN E: 3006-1466
ISSN P: 3006-1458
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
CJSSR
CONTEMPORARY JOURNAL OF SOCIAL SCIENCE REVIEW
Vol.04 No.01 (2026)

[5] R. Ahmed *et al.*, "AI-Driven Zero-Day Vulnerability Detection and Exploit Prediction," *International Journal of Computer Science and Engineering (IJCSE)*, 2025.
[6] Y. Zhao *et al.*, "Towards Deep Learning-Based Cybersecurity Risk Assessment," *Cluster Computing*, Springer, 2025, doi: 10.1007/s10586-024-05092-0.
[7] P. Singh *et al.*, "Enhancing Cybersecurity Vulnerability Detection Using Machine Learning Severity Models," *Gigvvy Science – International Journal of Applied Science and Engineering*, 2025.
[8] T. Müller *et al.*, "Machine Learning for Cybersecurity: Survey and Challenges," *Electronics*, vol. 14, no. 23, 2025, doi: 10.3390/electronics14234563.
[9] S. Ahmed *et al.*, "AI-Enhanced Cybersecurity Vulnerability Prevention," *Operations Research and Decision Making*, vol. 5, no. 1, 2024, doi: 10.56038/oprd.v5i1.616.
[10] Y. Li *et al.*, "Software Vulnerability Prediction: A Systematic Mapping Study," *Information and Software Technology*, vol. 165, 2024, Art. no. 107303, doi: 10.1016/j.infsof.2023.107303.
[11] R. Sommer *et al.*, "Vulnerabilities in Machine Learning for Cybersecurity Applications," *Journal of Information Security and Applications*, vol. 78, 2025, Art. no. 104269, doi: 10.1016/j.jisa.2025.104269.
[12] H. Wang *et al.*, "Automated CVE Severity Prediction Using Deep Learning and Explainable AI," *Research Square*, 2025, doi: 10.21203/rs-7123186/v1.
[13] K. Müller *et al.*, "Knowledge-Driven Large Language Models for Cybersecurity Automation," in *Proc. ACM Conf. on Computer and Communications Security (CCS)*, 2025, doi: 10.1145/3733817.3762699.
[14] L. Allodi *et al.*, "Exploiting the Known: Predicting Real-World Exploitation Using EPSS," *arXiv preprint*, arXiv:2302.14172, 2023. [Online]. Available: https://arxiv.org/abs/2302.14172
[15] X. Chen *et al.*, "CVE-Driven Attack Technique Prediction Using Domain-Specific Language Models," *arXiv preprint*, arXiv:2309.02785, 2023. [Online]. Available: https://arxiv.org/abs/2309.02785
[16] M. Khan *et al.*, "The Role of Machine Learning in Predicting Zero-Day Vulnerabilities," *International Journal of Scientific Research and Analysis*, vol. 10, no. 1, pp. 45–55, 2023, doi: 10.30574/ijsra.2023.10.1.0838.
[17] J. Sabottke *et al.*, "Exploitation of Software Vulnerabilities Using Topic-Based Machine Learning," *Information*, vol. 14, no. 7, 2023, doi: 10.3390/info14070403.
[18] S. Park *et al.*, "A Cyber Risk Prediction Model Using CVE Data," *Expert Systems with Applications*, vol. 229, 2023, Art. no. 121599, doi: 10.1016/j.eswa.2023.121599.
[19] R. Ahmed *et al.*, "AI-Driven Zero-Day Detection in Networked Systems," *International Journal of Computer Science and Engineering (IJCSE)*, 2025.
[20] A. Ankalaki *et al.*, "Cyber Attack Prediction: From Traditional Methods to Artificial Intelligence," Flinders University Research Repository, 2025.