Vulnerability Patch Prediction using LLM based Bert Model with Trustworthy AI Practice for Cyber Security Enhancement

Nihala Basheer [1], Shareeful Islam[1,2], Spyridon Papastergiou[2,3] , Haralambos Mouratidis[4] , Papagiannopoulos Nikolaos[5]

[1] School of Computing and Information Science
Anglia Ruskin University, East Road, Cambridge, UK
[2] Research and Innovation, MAGGIOLI S.P.A., Italy
[3]Department of Informatics, University of Piraeus, Greece
[4]Institute for Analytics and Data Science, University of Essex, UK
[5] Head, Digital & Innovation Services, Athens International Airport, Greece
{nihala.basheer, shareeful.islam}@aru.ac.uk,
spyros.papastergiou@maggioli.gr, h.mouratidis@essex.ac.uk,
papagiannopn@aia.gr}

Abstract. The regular update of the security vulnerabilities is crucial for an organization to mitigate the possibilities of their potential exploitation that can pose for cyber-attack. Despite their importance, timely updates are not always guaranteed, and many vulnerabilities remain unpatched for extended period of time which increases any security risk to the organizations. Organizations generally update patches manually, which introduces delays towards mitigation of potential exploitation and require huge effort and resources. In this context, we propose a novel approach that uses Large Language Model (LLM)-based CodeBERT model to predict the availability of an update or a patch relevant for the vulnerabilities. The approach adopts key trustworthy AI characteristics, including biasness and explainability, to operationalize trustworthy AI practice for the LLM-based CodeBERT model. The work has been evaluated on a real-world use case scenario from Athens International Airport to demonstrate the applicability of the approach through a test environment that emulates the airport's critical operating systems. Assets from key systems such as flight information display and access control have been considered in airports and associated with vulnerabilities. The results from the study show that the update is predicated for the key vulnerabilities such as CVE-2017-8464 and CVE-2020-1472 which link with Windows 7-based access control system and Oracle-based AODB database server of the use case scenario respectively. Also, model explainability is improved by the feature importance using SHAP and correlation using Heatmap technique. The key features for the model decision making are exploitability_score, epss, and attack_complexity. Truswortty AI practice is also operationalized through bias mitigating techniques such as class balancing and equalized odds to ensure fair and balanced training of the model.

Keywords: CodeBERT, Trustworthy AI, Explainable AI, Cybersecurity, Bias, Large Language Model, Vulnerability, Patch, Asset

1 Introduction

Vulnerability patches are critical for the entire digital infrastructure to ensure security and resilience. The cyber threats landscape is continuously evolving with sophisticated attack vectors exploited by these known vulnerabilities. The timely availability of patches can tackle the possible exploitation of vulnerabilities by malicious actors [1]. These patches improve cybersecurity postures by reducing the risk related to outdated or unpatched software. Despite its importance, it is still challenging to update the patch regularly, as there is a large number of vulnerabilities published each day [2]. Moreover, there is a lack of systems that predicate the update of the relevant vulnerabilities automatically. Therefore, organizations often must be in a reactive stance since it leaves the systems vulnerable for longer durations before patches are released. In this context, AI-based patch update systems can contribute to such contexts for effective management of vulnerabilities. However, AI models generally have a black-box nature, which requires a need to ensure the model explains the outcomes and reduces any bias while predicting the vulnerabilities.

In this respect, this work combines CodeBERT with Trustworthy AI to predict whether relevant vulnerability updates are available. CodeBERT is a transformer-based model trained on large-scale code repositories that can analyze historical vulnerability data, security advisories, and patch release trends to estimate when a vulnerability is likely to be patched; thus, proactive security management. The key contributions of this paper are summarized below:

- First, we propose a LLM-based CodeBERT architecture for predicting the availability of vulnerability patches to enable organizations to take proactive and informed decisions in mitigating security threats related with the vulnerabilities. CodeBERT can predict the availability of future patches by analyzing relevant vulnerability-related data, thus reducing the window of exposure concerning critical vulnerabilities. It follows a sequential phase that commences with data tokenization and model performance evaluation with trustworthy AI practice.
- Secondly, the novel proposed approach includes Trustworthy AI (T-AI) with a focus on bias mitigation and explainability characteristics towards T-AI practice [3]. As the AI-driven security models may inherit the biases from the historical data, it becomes of prime importance to make sure that the proposed LLM-based CodeBERT model for vulnerability patch prediction will be fair and unbiased through class balancing and equalized odds. Furthermore, we integrate explanation techniques like SHAP and heatmaps, which provide capabilities to identify key contribution features and correlation among them. Our approach enables fairness and interpretability, following the principles of Trustworthy AI, whereby organizations have better vulnerability management workflows with much-needed interpretability and transparency.
- Finally, a real case study from Athens International Airport is considered to demonstrate the applicability of the proposed approach. Data from this case study is linked with the vulnerabilities dataset through an experiment using the CVEjoin dataset [4]. This experiment focuses on the prediction of the availability of patches for related vulnerabilities affecting critical infrastructure systems. The result from the study shows that an update is predicted for important vulnerabilities such as CVE-2017-8464 and CVE-2019-0708 linked with the Windows 7-based access control system of the airport. Moreover, UFIS/AMADEUS applications are affected by CVE-2017-0037, while flight information integrity is affected by CVE-2020-0674.

2 Related Work

2.1 Necessity of Vulnerability Updates

Vulnerability patches serve the purpose of reducing cyber threats by plugging in loopholes before a malicious entity can take advantage of it. To stand against cyber threats, organizations need to have active cyber defense mechanisms and remove potential vulnerabilities that can lead to breaches in the future. Research has shown that effective vulnerability management requires more than just fixing existing problems. It also necessitates effective planning when dealing with new threats so that costs and impact on brand image are reduced, all while making sure systems are secure and protected [5]. Studies on the dependency management of open-source software and some strategies, such as the silent vulnerability fix, are performed without any prior communication with the customers or stakeholders, which puts their organization at risk [6]. Furthermore, considering a vulnerable database such as the NVD or MITRE means waiting endlessly for updates and patches, which creates a need for automated patching systems, fully eliminating manual efforts and ensuing security threats. These systems, along with being efficient, are also effective in shrinking the inverse of the attack surface as everything is dealt with in real time. Exploiting the long-term analysis of software vulnerabilities, many businesses unfortunately continue operating with obsolete and dangerously vulnerable software when more secure patches have since arisen.

According to recent research, the adoption of patches is influenced by cloud hosting and update complexity, which points to the need for more sophisticated policy measures and automation to improve responsiveness to updates [7]. There remain issues with implementing secure update mechanisms, especially since so many systems depend on trusted networks for critical patches that are subject to man-in-the-middle. Studies suggest that for effective unauthorized modification protection, there is a need for comprehensive, secure update protocols that utilize authentication, encryption, and digital signatures [8]. The patching of vulnerabilities, automation, tactical risk management, and stringent security measures make it possible to enhance cybersecurity resilience and reduce the chances of modern exploits in digital infrastructure.

2.2 Adoption of Trustworthy AI

The adoption of trustworthy AI is necessary to achieve equity, clarity, responsibility, and safety in AI-powered applications or systems. It is particularly important to establish trust in AI systems when they are deployed in more sensitive areas, including cybersecurity and risk assessment, to reduce biases, improve explainability, and alleviate risks. Various works have focused on various components of trustworthy AI, focusing on the incorporation of policy and ethics as well as cross-disciplinary approaches. One of the works [3], in particular, concerns AI models that are used in vulnerability detection, which focuses on BERT-based LLMs and explains why trustworthy AI is important. Such models attempt to use AI to solve the identification problem while avoiding the black-box problem. The study posits that by using explainability techniques, the model enables interpretable cybersecurity vulnerability management that complies with the transparency requirements of the EU AI Act. Likewise, another study [9] on the development of trustworthy AI has a more systematic approach and considers all phases of the AI life cycle, including data intake and post-deployment monitoring, as critical phases where trust must be established. It also argues for the inclusion of active protection against adversary actions and unintentional biases through explainability, fairness, privacy, and accountability.

In the context of LLMs, research [10] points out trust issues concerning bias, lack of transparency, and security risks which require ethical governance, requisite regulatory conditions, and responsibility from the industry. The research advances a Trust in AI assessment framework based on explainability, robustness, privacy, and governance to facilitate cooperation between technology, ethics, and policy. On the other hand, in another study [11], the analysis of the AI Act shows how the interaction between trust and risk acceptability biases AI governance. The argument is made that building trust purely through compliance is not acceptable and that continuous transparency, accountability, and engagement provide assurance about AI systems being trustable instead of legal-bound AI systems.

3 Proposed Approach

The proposed approach focuses on predicting whether a patch or update is available for vulnerabilities and performs the prioritization based on the EPSS score of the vulnerability. The different phases of the approach are discussed in the following sections, outlining the methodology for model training, performance evaluation and trustworthy AI practice.

3.1 Trustworthy AI Characteristics

AI systems have become an integral part of decision-making in many domains, but two major concerns arise regarding their reliability and fairness. Two major challenges with ethics in AI include bias and explainability. The mitigation of these challenges involves the development of AI systems that can be trusted to be fair, transparent, and interpretable.

- **Bias** in AI are systematic errors or unfair preferences in AI systems that may be against a group or individual. This may be the case when historical prejudices are embedded in training data, data collection is biased, or the model design itself contains problematic assumptions [12]. If biases are not handled properly within AI systems, they might just perpetuate or even scale existing social inequalities. For example, bias in healthcare AI systems can lead to poorer care for certain demographic groups and lending algorithms rejecting qualified applicants from minorities. These create a vicious circle where the AI system enhances the prevailing bias in society and contributes to disparity.
- **Explainability** is about understanding how AI systems make their decisions. Most modern AI, especially large language models, are simply "black boxes" where often there is difficulty tracing how conclusions are reached in detail [13]. Without transparency, there is a major risk of mistakes, biases, or failures being buried deep inside the system unless some form of explainability has been built in. Neglecting explainability can also make the application of artificial intelligence systems risky

in critical settings [13]. This lack of transparency can also blur the line into a lack of interpretability when AI systems make harmful decisions.

3.2 The proposed vulnerability patch prediction

Figure 1 presents the architecture of the proposed approach, specifically designed to uphold Trustworthy AI (T-AI) practices with CodeBERT model with three sequential structure phases.

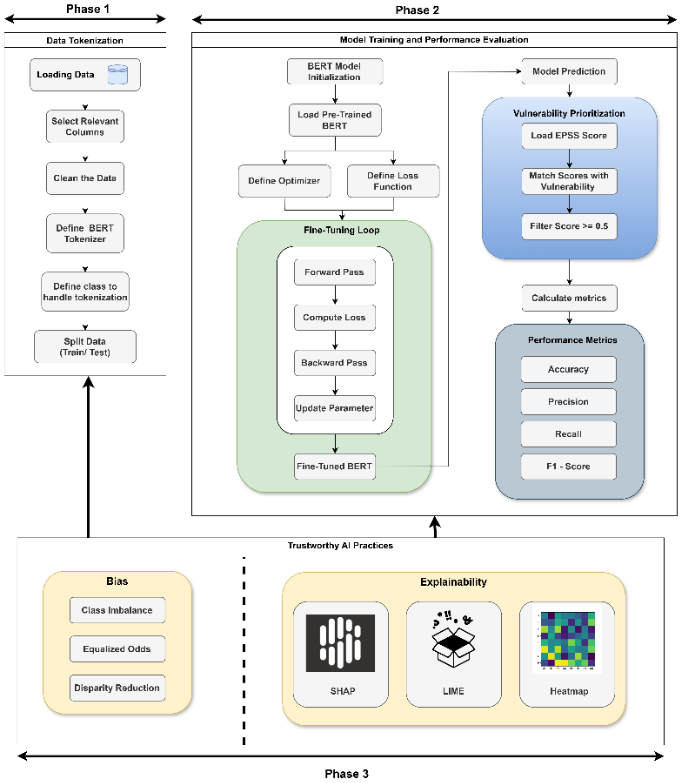

Fig. 1. Proposed Model

- **Phase 1 (Data Tokenization):** The initial phase starts with selecting the columns relevant for the task and later cleaning them from inconsistencies and missing values. When quality in data is ensured, it loads a pre-trained BERT tokenizer using the Hugging Face Transformers library. It selects the text columns for processing: 'description', 'vendor', and 'product'. In this step, these were combined into one text string for each row in the dataset. Then, tokenization with padding and truncation is applied to the combined text to have all samples of equal sequence length. Finally, results of tokenized text are inserted as input IDs, and a new column named 'tokenized_text' is stored in the dataset.
- **Phase 2 (Model Training and Performance Evaluation):** This phase focuses on fine-tuning the CodeBERT-based LLM model so that it can predict the available patch for a certain vulnerability. Initially, the pre-trained BERT model is loaded,

and then the model is trained for this particular binary classification task, which has two output labels. AdamW is the contemporary optimizer for almost all transformer-based models, especially for training in GPU environments [14]. There is cross-entropy loss because the task is classification. The training loop iterates over the data for five epochs. For every iteration, the tokenized inputs and attention masks will be obtained from the model. Predictions will be made, from which the loss will be calculated. After the loss has been determined, the average total loss over each epoch is utilized in order to monitor performance. It then computes gradients for backpropagation, updating the model parameters using the defined optimizer. The effectiveness of the BERT-based model will be assessed with multiple metrics scores, such as accuracy, precision, recall and F1 score [15]. Additionally, we make use of the EPSS score for prioritizing vulnerabilities. We place priority on the vulnerabilities that have an EPSS score above 0.6 because they are very likely to be exploited. This strategy enables organizations to concentrate resources on applying updates and patches where it is most needed, which improves the overall security of the system.

- **Phase 3 (Trustworthy AI Practice):** The final phase emphasizes the practices of T-AI that uphold fairness and transparency in the decision-making process on the prediction and availability of vulnerability patches. T-AI improves a model's reliability, security, and accountability, which in turn enhances its effectiveness in real-world scenarios. In this work, we limit our scope to bias and explainability characteristics. This is because both bias and explainability stem from the central theme of equal diverse group representation and, more importantly, the need to offer assurance to stakeholders' trust through provision of explainable models.
  - **Bias** mitigation is important to undertake to ensure that AI systems are ethical and that they are fair, especially within the context of vendors, products, and all types of vulnerability. Equalized odds are considered to guarantee that the model will achieve equal effectiveness across all specified groups or categories through reduction of any performance gaps that could result in biased outcomes. Class balancing balances the over-representation of common vulnerabilities by giving equal weight to rare, critical vulnerabilities during training so that the model does not get biased toward frequently occurring cases. These techniques taken together make the model's predictions fairer and more equitable.
  - **Explainability** is utilized to enhance the interpretability, trust, and understanding of the inner workings of a model among the users through techniques such as SHAP and heatmap. This will enable the users to understand what features and factors determine the models' decisions for its predictions. For instance, SHAP can be applied to explain the importance of features on a global as well as local level for specific predictions. Moreover, heatmaps provide correlations among features that provide dependencies between features that aid in determining factors that probably control the model. This level of interpretability is relevant for stakeholders to validate the model for fairness, identify potential biases, and make sure it conformed to the ethical standards set for AI.

4 Evaluation

The proposed approach is evaluated through a real case study from Athens International Airport. This implementation helps in forecasting the vulnerabilities which are anticipated to receive updates or patches while ensuring the T-AI practices. The objective of such evaluation is:

- To evaluate the applicability of the approach in real-life scenario.
- To assess the prediction of vulnerabilities which are expecting to receive updates or patches.

4.1 Industrial Use Case Scenario

The selected scenario is based on the Athens International Airport (AIA), the international airport serving the capital city of Greece, which is one of the busiest airports in Europe [16]. The airport incorporates a number of business key systems, including flight information display, passenger information system, and baggage handling business operations. In this paper we have considered only three services as a part of the test environment that emulates the airports critical operating systems:

- **Flight Information Display System (FIDS)** is a designated display system for monitoring and displaying important data related to specific flights and their details, such as passenger arrival or departure times from specific terminals along with notifications and information for important flight management variables.
- **Advance Passenger Information System (APIS)** provides essential communication functionalities for the collection and transmission of key information to be extracted from the Machine-Readable Zone (MRZ) of Machine-Readable Travel Documents (MRTDs) or Passports. These details are then sent to border control authorities before the flight departs or arrives so that it is already available at the primary inspection post when the traveller reaches the airport.
- **Access Control System (ACS)** is an important part of the information system security and border control and contains the functions of controlled access to a zone through an authentication device. It permits only those individuals who are recognized by an airport's restricted area access control list to enter such areas.

Assets and CPE related to the selected system: These chosen systems heavily rely on several assets which are listed below with the CPE (Common Platform Enumeration) ID [17]. This allows linking the asset with the potential vulnerabilities.

- Operating system: Microsoft Windows 7 based access control client with CPE ID cpe:2.3:o:microsoft:windows_7:-:*:*:*:*:*:x64:*, Microsoft Windows server based CCTV server andC CTV gateway with CPE ID cpe:2.3:o:microsoft:windows_server_2012:-:*:*:*:*:*:*:*.
- Airport Operational Database (AODB): Linux Server based AODB APP with CPE ID cpe:2.3:o:oracle:linux:5:-:*:*:*:*:*, Windows based AODB client with CPE ID cpe:2.3:o:microsoft:windows_7:-:*:*:*:*:*:x64:* , Oracle based AODB database server with CPE ID cpe:2.3:o:oracle:linux:5:-:*:*:*:*:*:*.
- Universal Flight Information System (UFIS): UFIS/AMADEUS based Client Application and database server with suggested CPE ID cpe:2.3:a:systematicinc:sitaware:6.4:sp2:*:*:*:*:*:*

- Data: staff credential records, passenger information

Attack Scenario: Spreading phishing e-mails to the airport's internal operators could amend the flight information and continue to affect other parts of the network. Attack Vector: malicious attachment through e-mail which triggers ransomware with types LockBit, or WannaCry.

4.2 Experiment

The experiment part of the evaluation considers the CVEjoin [5] data set to predicate vulnerabilities which are potential for update. Therefore, identified assets and their related CPE are considered to link the asset with the dataset so that appropriate CVE ID can be taken into consideration.

Dataset Description: The CVEjoin dataset represents a general set of 200,473 vulnerabilities, with enriched metadata concerning the classification of vulnerabilities, impact, exploitability, and patch availability provided by a large community of contributors. In particular, a CVE ID has been assigned to each vulnerability included in this dataset, and vulnerabilities have been categorized by using the CWE framework. We have identified the CPEs related to the assets of the use case scenario and mapped them with CWE based on the relevant CPE of the asset.

Key features:

- It includes the affected components of the vendor, product, and system type.
- Each vulnerability holds a CVSS score and a severity rating indicating how impactful the vulnerability is; in addition, the needed resources and type of attack grade all help quantify risks and impacts, allowing for assessment of each vulnerability.
- Security advisories and references indicate if an update or patch is available.
- Threat intelligence indicators included in the dataset are the Exploit Prediction Scoring System (EPSS) and Google trends.

4.3 Results

The results obtained through the experiment show that an effective prediction of vulnerabilities regarding patch availability based on their EPSS scores and their contributions to enhance security of the Athens Airport.

Data tokenization: After cleaning the dataset and selecting relevant columns, tokenization was performed as outlined in Section 3.2, utilizing the pre-trained BERT tokenizer from the Hugging Face Transformers library. The tokenization process generated, among other outputs, input IDs and attention masks, all of which will be used by the BERT-based model.

Model training and Performance evaluation: Table 1 depicts the performance of the model in 5 epochs, with the accuracy improving from 0.9651 to 0.9692 and the training loss decreasing from 0.1206 to 0.1006.

Table 1. Performance Metrics Evaluation

No of Epochs	Accuracy	Training Loss	Precision	Recall	F1 - Score
1	0.9651	0.1206	0.5932	0.0629	0.1137

2	0.9682	0.1067	0.7100	0.0638	0.1171
3	0.9690	0.1037	0.7129	0.0647	0.1186
4	0.9690	0.1020	0.7934	0.0863	0.1556
5	0.9692	0.1006	0.7557	0.0889	0.1592

Lastly, after performance evaluation, prioritizing the vulnerabilities will be done with the EPSS score. This prioritization is critical for focusing on the most pressing vulnerabilities. The vulnerabilities are filtered and ranked according to their EPSS scores. Particularly, vulnerabilities with an EPSS score greater than 0.6 will be prioritized since these suggest a greater scope of exploitation. The table below contains the prioritized vulnerabilities.

Table 2. Vulnerability Prioritization using EPSS Score

CWE	Vendor	Product Name	EPSS Score
CVE-2017-8464	Microsoft	‘windows_server_2016', 'windows_10', 'windows_rt_8.1', 'windows_8.1', 'windows_7', 'windows_server_2008/2012'	0.96393
CVE-2019-0708	Microsoft	'windows_7', 'windows_xp/vista', 'windows_server_2003/2008',	0.96235
CVE-2017-0037	Microsoft	'edge', 'internet_explorer'	0.96089
CVE-2014-0160	Debian	‘debian_linux'	0.96076
CVE-2020-0674	Microsoft	'windows_10/8.1/7', 'internet_explorer', 'windows_rt_8.1', 'windows_server_2008/2012/2019'	0.95631
CVE-2020-1472	Oracle	'zfs_storage_appliance_kit	0.95011
CVE-2020-0618	Microsoft	'sql_server	0.94676
CVE-2018-4878	Linux	windows', 'chrome_os', 'flash_player', 'linux_kernel', 'macos	0.75301

TAI practices: For ensuring TAI practices, bias mitigation techniques such as class balancing and equalized odds were performed for nondiscriminatory training and avoiding disparities in vulnerability predictions. The explainability techniques used were SHAP and heatmaps for improving model transparency, providing insights into feature importance and decision-making processes.

- **Bias:** Figure 2 represents the class distribution analysis, showing a high imbalance in the classes before the SMOTE (Synthetic Minority Over-sampling Technique) application. Originally, there were 155,027 instances for Class 0 and 5,351 for Class 1, making the dataset imbalanced, with a high preponderance toward the majority class. By oversampling the minority class, SMOTE balances both classes to 155,027. This is important in balancing the model to ensure fair predictions and prevent bias towards the majority class.

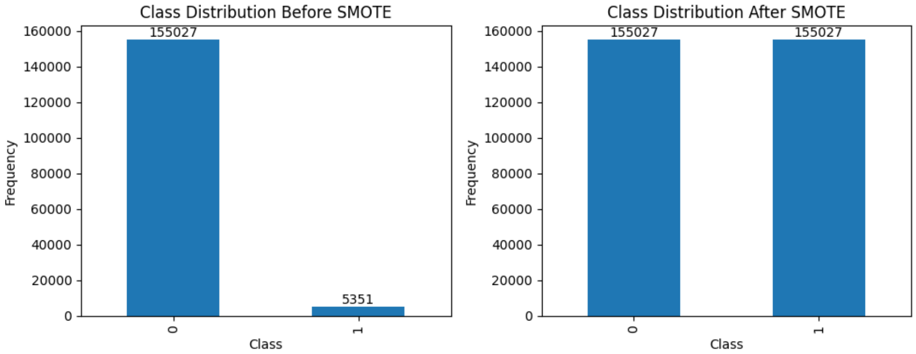

Fig. 2: Class balancing before and after SMOTE

The Equalized Odds findings as presented in Table 3 show that the model has uniform True Positive Rates (TPR) over all attack vectors, with figures between 0.7596 and 0.7667. This suggests that the model performs for vulnerability detection at fair rates over adjacent network, local, network, and physical attack vectors.

Table 3: Equalized Odds result

Attack Vector	TPR	FPR
Adjacent Network	0.766667	0.001131
Local	0.763382	0.001824
Network	0.759577	0.004098
Physical	0.762343	0.000897

- **Explainability:** Figure 3 shows the SHAP result, displaying the average impact of each feature on the model output, highlighting the relative importance. The most influencing factor is the ‘exploitability_score’, having the highest mean SHAP value, hence highly affecting the model's predictions. This is followed by ‘epss’, which also plays a major role in shaping the model's decisions. Other important features such as ‘confidentiality_impact’, ‘attack_vector’ and ‘impact_score’ have a middle-level influence on the model outcome. While ‘privileges_required’, ‘user_interaction’, and scope are on the weaker side. However, some features like ‘availability_impact’, ‘base_score’, ‘integrity_impact’, and ‘attack_complexity’ are considered least important, as their contribution to the model’s predictions is low.

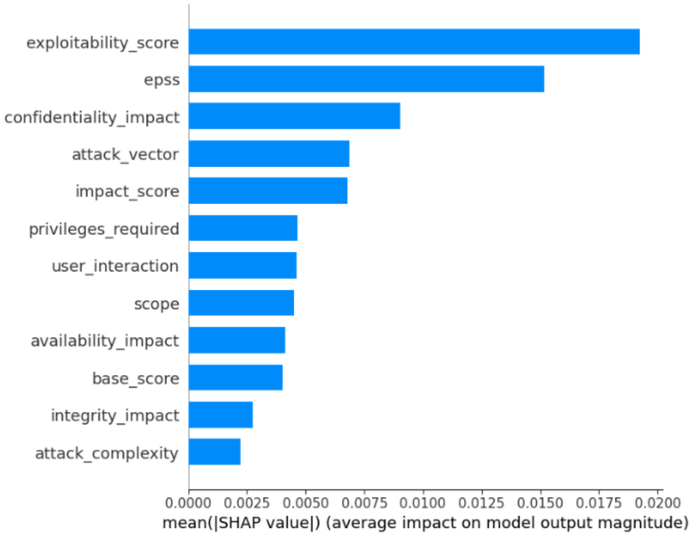

Fig. 3: Global feature importance using SHAP

Figure 4 shows the impact of the local importance of the analyzed features on the prediction of the model. The base value is the expected model output before feature influences, which is approximately 0.01791, while the final model output equals 0.05. In addition, some features such as 'core', 'scope', 'attack_vector', 'confidentiality_impact', and 'exploitability_score' are the features which drive this prediction high. Of these, the most positively influential feature is 'exploitability_score'. On the contrary, 'epss' and 'attack_complexity' make for a deduction in the predicted value, for which ‘epss’ has the greatest negative impact. The final prediction is thus created by the balance of these phenomena: while all attack-related features increase the perceptions of risk, complexity and the exploit prediction score reduce it.

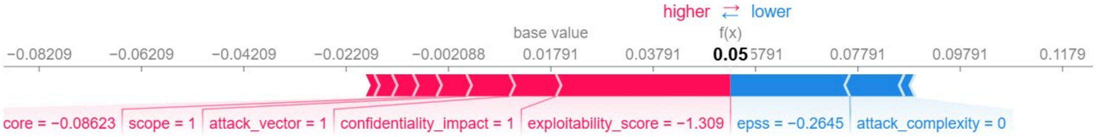

Fig. 4. Local feature importance using SHAP

Figure 5 represents a heatmap for the correlation between chosen features of a vulnerability, namely product, vendor, attack_vector, epss, update_available, base_score, exploitability_score, impact_score, integrity_impact, and confidentiality_impact. The range of color goes from red, which shows positive correlation. to blue, which shows negative correlation, while numeric notations depict how strong those relationships are. This instrument makes it possible to point out primary correlations between vulnerability characteristics.

- The strong positive relations between the features impact_score, base_score, and confidentiality_impact are colored in red.

- While weaker relationships - specified by blue - exist between categorical variables such as vendor and attack_vector.

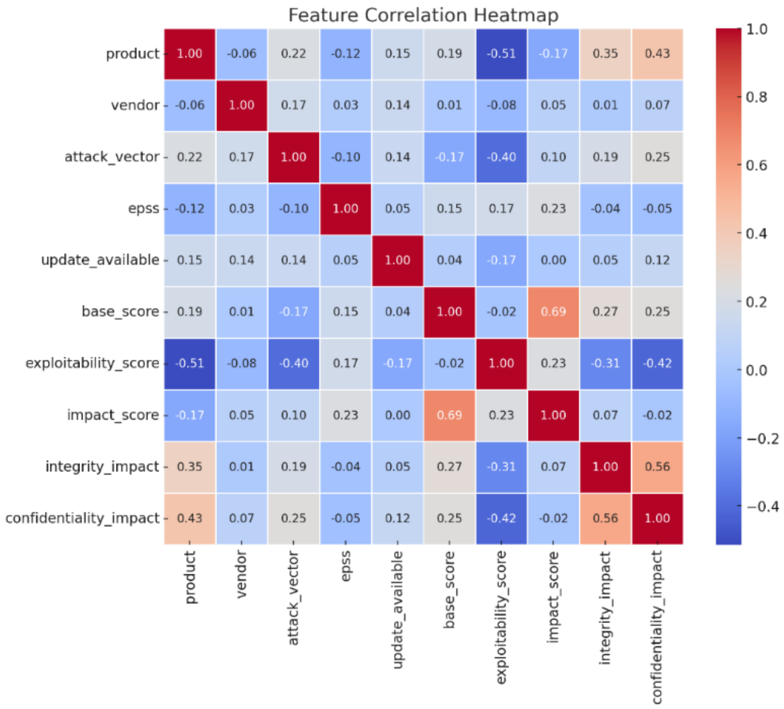

Fig. 5. Feature correlation Heatmap

4.4 Discussion

Patch updates are essential for managing vulnerabilities. They help in mitigating the risk of a breach which can lead to unauthorized access, the execution of harmful code, system disruptions, loss of sensitive data, and financial troubles [18]. Patching and prioritizing resource allocation in a business can also be done through the EPSS, which helps reduce the overall risk to security. This work aims to predict the availability of patches or updates for the relevant vulnerabilities that follow a systematic architecture which includes T-AI practices. Moreover, this approach has an emphasis on an EPSS scoring system that focuses on vulnerabilities with 0.6 and above to help mitigate the vulnerabilities that are most likely to be exploited. The prediction model is improved with the CodeBERT model, which is known for its strong understanding of context in text data. Since BERT is bidirectional, it takes into account the context surrounding a sentence, which assists in posing a classification of whether a patch is available or not [19].

In the context of T-AI, we have considered bias and explainability, which ensures the fairness and interpretability of the decision-making of the model. Addressing bias ensures that different groups have equal chances to benefit and do not suffer from disadvantages arising from such unfair practices or policies [12]. The equalized odds and class imbalance handling techniques allow us to formulate a fairer model which otherwise might result in very extreme inequitable benchmarks. Moreover, explainability makes it easier for stakeholders or users to understand and trust the model's predictions because the rationale behind the model predictions and decision-making are clear [13]. These characteristics together guarantee trustworthy operation of the AI system, fulfil ethical requirements, comply with the regulations, and enable accurate vulnerability patch prediction.

The result from the case study demonstrates that the proposed methodology provides the capabilities to predict the possible update to the related vulnerabilities. The result from the use case scenario demonstrates that key assets such as Flight Information Display System or Access Control System link with CVEs such as CVE-2017-8464, CVE-2019-0708 or CVE-2017-0037 which are potential to exploit. Therefore, it is necessary to identify the update patches so that the vulnerabilities can be controlled, and AIA can avoid any potential security risk that can pose potential disruption to their critical services.

6. Conclusion

In this day and age, where cyber threats are becoming increasingly rampant, the need for timely and effective vulnerability patches are of utmost importance. This paper proposes a novel approach that predicts the availability of vulnerability patches by utilizing the capabilities of the CodeBERT LLM model, along with T-AI practices. The systematic approach, which includes data tokenization and performance evaluation, ensures that the model is reliable and robust. The integration of T-AI is aimed at mitigating bias and explaining the AI-driven security system. This not only shifts the security paradigm but enhances the transparency and interpretability of security powered by AI systems. The proposed model was examined and effectively applied to a use case scenario that considers the key systems of Athens International Airport, which are validated using the CVEjoin dataset. The findings categorize the model as proficient in estimating the patch timelines for critical infrastructure vulnerabilities. In our future work, we would like to incorporate additional characteristics into the T-AI practices and evaluate the findings with various datasets and application cases for broader integration.

Acknowledgement
This work was supported by the European Union's Horizon Europe Project, CUSTODES - A Certification approach for dynamic, agile and reUSable assessmenT fOr composite systems of ICT proDucts, servicEs, and processes, funded under grant agreement No. 101120684 and CyberSecDome — An innovative Virtual Reality based intrusion detection, incident investigation and response approach for enhancing the resilience, security, privacy and accountability of complex and heterogeneous digital systems and infrastructures, funded under grant agreement No. 101120779.

References

1. Xie, Z., Wen, M., Wei, Z., & Jin, H. (2024). Unveiling the Characteristics and Impact of Security Patch Evolution. ASE '24: Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering, 1094–1106. https://doi.org/10.1145/3691620.3695488

2. Islam, S., Abba, A., Ismail, U., Mouratidis, H., & Papastergiou, S. (2022b). Vulnerability prediction for secure healthcare supply chain service delivery. Integrated Computer-Aided Engineering, 29(4), 389–409. https://doi.org/10.3233/ica-220689
3. Haurogné, J., Basheer, N., & Islam, S. (2024). Vulnerability Detection using BERT based LLM Model with Transparency Obligation Practice Towards Trustworthy AI. Machine Learning With Applications, 18, 100598. https://doi.org/10.1016/j.mlwa.2024.100598
4. figshare. (2022, November 24). CVEJoin: an Information Security Vulnerability and Threat intelligence dataset. Figshare. https://figshare.com/articles/dataset/CVEjoin_A_Security_Dataset_of_Vulnerability_and_Threat_Intelligence_Information/21586923?file=38314677
5. Rantalaiho, V. (2024). Technical implementation and operational enhancements of a vulnerability management tool in an organization. Theseus. https://www.theseus.fi/handle/10024/851234
6. J. Sun et al., "Silent Vulnerable Dependency Alert Prediction with Vulnerability Key Aspect Explanation," 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), Melbourne, Australia, 2023, pp. 970-982, doi: 10.1109/ICSE48619.2023.00089.
7. Murciano-Goroff, R., Zhuo, R., & Greenstein, S. (2024). Navigating Software Vulnerabilities: Eighteen Years of Evidence from Medium and Large U.S. Organizations. https://doi.org/10.3386/w32696
8. Murciano-Goroff, R., Zhuo, R., & Greenstein, S. (2024). Navigating Software Vulnerabilities: Eighteen Years of Evidence from Medium and Large U.S. Organizations. https://doi.org/10.3386/w32696
9. Li, B., Qi, P., Liu, B., Di, S., Liu, J., Pei, J., Yi, J., & Zhou, B. (2022). Trustworthy AI: From principles to practices. ACM Computing Surveys, 55(9), 1–46. https://doi.org/10.1145/3555803
10. Ferdaus, M. M., Abdelguerfi, M., Ioup, E., Niles, K. N., Pathak, K., & Sloan, S. (2024). Towards Trustworthy AI: A review of ethical and robust large language models. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2407.13934
11. Laux, J., Wachter, S., & Mittelstadt, B. (2023). Trustworthy artificial intelligence and the European Union AI act: On the conflation of trustworthiness and acceptability of risk. Regulation & Governance, 18(1), 3–32. https://doi.org/10.1111/rego.12512
12. Kokate, N. A., & Priya, N. M. (2024). Bias and its Consequences: A study of Machine Learning performance. International Journal of Scientific Research in Computer Science Engineering and Information Technology, 10(6), 290–301. https://doi.org/10.32628/cseit241051088
13. Inukonda, J., Tetala, V. R. R., & Hallur, J. (2024). Explainable Artificial intelligence (XAI) in healthcare: Enhancing transparency and trust. International Journal for Multidisciplinary Research, 6(6). https://doi.org/10.36948/ijfmr.2024.v06i06.30010
14. Graetz, F. M. (2020, February 12). Why AdamW matters - Towards Data Science - Medium. Medium. https://medium.com/towards-data-science/why-adamw-matters-736223f31b5d
15. Su, J., & Wu, Y. (2024). Refining CVE-to-CWE mapping with enhanced attention in BERT-based models. Applied and Computational Engineering, 71(1), 107–112. https://doi.org/10.54254/2755-2721/71/20241647
16. Travellers. (n.d.). Athens Internation Airport. https://www.aia.gr/en
17. NVD - Search. (n.d.). https://nvd.nist.gov/products/cpe/search
18. Insurance, P. (2023, October 26). Risks of unpatched vulnerabilities | ProWriters Insurance. ProWriters. https://prowritersins.com/cyber-insurance-blog/unpatched-vulnerability-risks/
19. Eang, C., & Lee, S. (2024). Improving the accuracy and effectiveness of text classification based on the integration of the BERT model and a recurrent neural network (RNN_BerT_Based). Applied Sciences, 14(18), 8388. https://doi.org/10.3390/app14188388