Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.5567

Portfolio-based Active Learning with Gaussian Processes for Vulnerabilities Risk Classification

Davyson S. Ribeiro [**Federal University of Ceará** | ***davysonribeiro@alu.ufc.br***]
Rafael S. Lemos [**Federal University of Ceará** | ***rafael.lemos@alu.ufc.br***]
Francisco R. P. da Ponte [**Federal University of Ceará** | ***fco.rparente@gmail.com***]
César Lincoln C. Mattos [**Federal University of Ceará** | ***cesarlincoln@dc.ufc.br***]
Emanuel B. Rodrigues [**Federal University of Ceará** | ***emanuel@dc.ufc.br***]

Federal University of Ceara (UFC), University Avenue, 2853 - CEP 60020-181 - Fortaleza - CE - Brazil

Received: 15 February 2025 • **Accepted:** 28 June 2025 • **Published:** 02 March 2026

Abstract. Effective vulnerability management is essential for cybersecurity, particularly as the demand for skilled professionals often exceeds supply. This paper investigates the application of Gaussian Processes (GPs) integrated with Active Learning (AL) techniques to classify security vulnerabilities based on their risk of exploitation. The main objective is to optimize the labeling process, thereby reducing the amount of labeled data necessary for training an effective classifier. The proposed methodology combines the uncertainty predictions provided by GP models with five established data selection strategies, utilizing a portfolio-based approach. The portfolio avoids the need of choosing a single strategy and leverages the strengths of each technique. This approach enhances adaptability and balances exploration versus exploitation in complex optimization scenarios, ultimately improving the diversity of labeled samples and contributing to the development of better classifiers trained with less examples. Experiments were conducted using the CVEjoin dataset, which encompasses over 200,000 vulnerabilities, across three distinct evaluation scenarios. The different setups consider equivalent volumes of labeled data, but varying Active Learning iterations. When considering a single strategy, the results indicate that the BSB (best and second best) method consistently outperformed the others in terms of accuracy and F1 score, particularly with an increased number of labeling iterations. In the scenario where multiple strategies are used in a portfolio, the results indicate gains in all evaluation metrics. This study underscores the usefulness of a portfolio-based Active Learning approach in optimizing the labeling procedure and, ultimately, prioritizing vulnerabilities for remediation. This research lays the groundwork for extending the framework to other areas of cybersecurity, such as vulnerabilities in web applications and cloud environments, thereby improving overall security measures in the digital landscape.

Keywords: Vulnerability Risk Classification, Machine Learning, Cybersecurity, Active Learning, Gaussian Processes.

1 Introduction

The growing number of connected devices in corporate networks and critical systems makes their management increasingly challenging. This also broadens the attack surface, exposing computational systems to malicious users and requiring robust security strategies to mitigate vulnerabilities [Jakkal, 2022]. The National Institute of Standards and Technology (NIST) defines vulnerabilities as weaknesses in an information system or its implementation that can be exploited by criminals [Ross, 2012].

An effective way to address this problem is to proactively address vulnerabilities before they can be exploited. This approach is called Vulnerability Management (VM), an ongoing process that identifies, classifies, prioritizes, and remediates flaws in software, applications, and operating systems [Foreman, 2019]. To ensure a secure computing environment, it is essential to assess vulnerabilities based on their severity and likelihood of exploitation [Sabottke *et al*., 2015]. This is done by collecting data from multiple sources, including Common Vulnerabilities and Exposures (CVEs) registered in the National Vulnerability Database (NVD) [1], threat intelligence feeds, security advisories, and vendor bulletins, which provide up-to-date information on emerging threats, helping to understand the associated risks and how to combat them.

However, analyzing the vast volume and variety of data collected in a computing environment also presents a significant challenge for security teams. In addition to leveraging NVD data on CVEs, it is critical to consider contextual information specific to an organization's computing environment—such as asset relevance and operational importance. Collecting this information enables a more accurate risk assessment, providing insights into the likelihood of exploitation of vulnerabilities within the organizational context [Tenable, 2023]. By integrating technical and contextual factors, security teams can more effectively identify and prioritize the vulnerabilities that pose the greatest risk to the organization.

The number of security professionals available to handle this type of task is small relative to the number of systems and vulnerabilities that require attention [Hore *et al*., 2023]. Identifying and classifying flaws in critical components, such as networks, applications, and operating systems, is a com-

[1] National Vulnerability Database, `https://nvd.nist.gov/`

plex and time-consuming process, often performed manually. This task requires specialized technical knowledge, making it even more challenging. Accurate labeling of vulnerabilities, especially regarding the risk of exploitation, is essential to prioritize the most urgent threats and perform the necessary remediations. With effective classification, security teams can optimize their efforts by focusing on proactive mitigation strategies and defense mechanisms instead of relying on manual classification [da Ponte *et al*., 2023].

Given the need to prioritize vulnerabilities and the limited resources of information security teams, Machine Learning (ML) techniques can be used to classify vulnerabilities based on risk [Alshaya *et al*., 2023]. Active Learning (AL), a subarea of ML, identifies the most informative examples for labeling, reducing the amount of data required for training and the evaluation effort. This process allows the classification model to efficiently prioritize vulnerabilities, simulating the expertise of information security experts and optimizing remediation efforts.

Vulnerability labeling is a challenging process, especially when available data is limited. Gaussian Process (GP) models emerge as a promising solution, as they are nonparametric Bayesian methods that are particularly effective in scenarios with few labeled data [Williams and Rasmussen, 2006]. Instead of providing only point estimates, a GP model provides probability distributions, allowing to quantify the uncertainty associated with predictions. This capability makes GPs ideal for Active Learning (AL), where the selection of the most informative samples is essential. By using GPs to identify the most uncertain samples, the proposed method reduces the need for large labeled datasets, allowing the model to focus on the most complex and ambiguous cases, which reduces the human effort in labeling and improves the efficiency of the process, while maintaining satisfactory generalization.

A limitation of traditional AL methods is that they often rely on a single sample selection strategy, which can result in imbalanced exploration of the data space. To overcome this limitation, we propose a portfolio-based approach (PA) that uses multiple selection strategies to guide decision making in complex optimization scenarios. The idea is to leverage the strengths of each strategy to achieve greater adaptability and balance exploration and exploitation [Vasconcelos *et al*., 2019]. Although this concept has been widely used in Bayesian optimization Hoffman *et al*. [2011], it can be easily adapted to AL. The use of multiple selection strategies improves the diversity of training samples, reducing bias and ensuring more representative coverage of the features of the analyzed patterns. This contributes to the robustness of the model, allowing it to explore under-explored areas while improving previously understood regions.

By integrating GP with AL, this work introduces an approach that reduces the need for large labeled datasets while improving classification performance. The portfolio-based selection strategy further enhances the process by offering a dynamic selection of the most informative samples. This approach optimizes the labeling effort and enables more efficient prioritization of vulnerabilities, which is essential for cybersecurity challenges. As the model refines its understanding of the data, it becomes more agile in adapting to new threats, ensuring robust risk classification with minimal human intervention.

This approach aims to improve the overall performance of the model by leveraging the advantages of each strategy to increase accuracy and reduce labeling effort, especially in scenarios with limited data. The classifier performance tends to improve over time as more informative examples are labeled and incorporated into the training set. This was validated by comparing the uncertainty metrics alone and the portfolio strategy, achieving 77% and 79% accuracy, respectively, compared to 74% for the random selection method.

Given the cybersecurity context, this work presents the following contributions:

1. Proposal for the use of GPs and AL for vulnerability risk classification, taking advantage of the GPs' ability to deal with uncertainties and improve the accuracy of predictions in the cybersecurity domain;
2. Evaluation of multiple AL strategies, identifying the most effective method for labeling vulnerabilities with minimal human effort, ensuring effectiveness in risk classification;
3. Evaluation of a portfolio-based sample selection approach, introducing a combination of strategies to optimize model efficiency and classification accuracy.

The rest of the paper is organized as follows: Section 2 presents concepts about AL and GP. Section 3 presentes a review of related reviews works related to ML and AL in the context of cybersecurity. Section 4 brings new updates on the portfolio-based approach for sample selection. Section 5 discusses the dataset used, the evaluation methodology and the main configurations used in the experimental scenarios related to AL and GP. Section 6 shows the results related to the evaluation of the selection strategies individually. Section 7 evaluates the approach using the portfolio of strategies, section 8 presents some limitations of this study. Finally, Section 9 presents the conclusion, summarizing the findings and possible future directions for research.

2 Active Learning

This section introduces the main concepts related to AL, treating it as a technique that seeks to optimize the model training process by reducing the amount of labeled data needed to achieve good performance. In ML problems, algorithms often require large volumes of labeled data to learn to make accurate predictions. However, as mentioned in the previous section, manually labeling data can be a costly and time-consuming task, especially in complex domains such as cybersecurity. This technique aims to minimize this effort by intelligently identifying which data examples should be labeled to improve the model more efficiently.

In AL, the machine learning model starts with a small set of labeled data and makes predictions for a large set of unlabeled data. The difference with AL is that, instead of training the model with a large volume of labeled data, the algorithm actively chooses the most informative data, that is, the examples that are more difficult to classify or that are on the border between different classes. These examples are

then sent to an oracle (an expert or automated system) to be labeled.

This process is iterative: In each round, the model is updated with the new labeled data, and from there it selects more samples for labeling. The objective is to optimize the time and resources necessary for labeling by reducing the number of labeled examples and enabling the model to learn more efficiently.

2.1 Types of Selection Strategies in Active Learning

There are several sample selection strategies in AL, each with their own benefits and limitations [Blasco *et al*., 2024]. The main selection strategies in the context of Active Learning are presented below.

- **Uncertainty Selection**: Is one of the most common strategies in active learning. In this approach, the model selects samples about which it has the greatest uncertainty, that is, those for which it does not have a clear or reliable classification. This uncertainty can be measured in several ways, such as the difference in probability between the most likely classes or the variance of the model's predictions. In classification tasks, the model selects samples whose predicted classes have probabilities similar to or close to the decision boundary, indicating greater uncertainty. In regression tasks, uncertainty is calculated based on the variance of the predictions, with the most uncertain samples having the highest variance.
- **Diversity Selection**: Seeks to select samples that cover the widest possible range of features or regions of the input space. The main goal is to ensure that the model is trained on a representative range of data, without focusing too much on a narrow subset. Selection is done based on the distance between samples, using metrics such as Euclidean distance or other forms of similarity. This type of selection ensures that the model learns about different aspects of the data, while avoiding overfitting on a homogeneous set of examples.
- **Representativeness Selection**: Focuses on choosing samples that are representative of the unlabeled data set as a whole. The model selects samples that are most typical or central within the input space, ensuring that the labeled data set is a good representation of the overall data. In many cases, this is done using clustering techniques such as K-means to select the central samples from each cluster, which allows the model to capture the overall characteristics of the data efficiently.
- **Margin Selection**: Is commonly used in classification tasks and is based on the difference in probability between the most likely classes. The idea is that the model chooses samples whose difference in probability between the two most likely classes is small, that is, those in which the model is most uncertain about which class is the most likely. This type of sample is located on the "border" between classes, representing cases that are more difficult to classify and therefore more informative for training.
- **Class Expansion Selection**: Or Query-by-Committee (QBC), involves building multiple models, each with a different hypothesis or configuration, for the same problem. The sample to be selected is the one in which the models disagree most strongly, that is, the samples that cause a large divergence in predictions. This is based on the idea that samples in which the models disagree indicate uncertainty and are therefore valuable for model learning.
- **Error-Expectancy Selection**: Focuses on identifying the samples that, when labeled, would result in the greatest reduction in model error. Rather than simply selecting uncertain or representative samples, the model chooses those that have the greatest potential impact on reducing classification or prediction error. The idea is that labeling these most impactful samples can lead to a substantial improvement in the model's overall performance.

Therefore, Table 1 presents the main advantages and disadvantages between the selection strategies presented in this section.

2.2 Gaussian Process Model

Gaussian Process Models provide analytical inferences mainly for Gaussian likelihood regression tasks [Rasmussen and Williams, 2006]. Since the problem addressed involves multiclass classification, a categorical likelihood implemented by the softmax function was chosen.

Following sparse GPs and variational inference [Hensman *et al*., 2013, 2015] strategies, model predictions are approximated by Monte Carlo samples, which are passed to the softmax function to result in probabilities. Then, the average probability of each class can be calculated by taking the average of the generated samples. More specifically, considering an input $\boldsymbol{x}_*$, representing an unlabeled vulnerability, and the corresponding s-th Monte Carlo sample $f_c^{(s)}(\boldsymbol{x}_*)$ of the model posteriori for the risk class c, the probability of the outcome (predicted class) y_* will be given by:

$$P(y_* = c \mid \boldsymbol{x}_*) = \frac{1}{S} \sum_s \text{softmax}(f_c^{(s)}(\boldsymbol{x}_*)), \quad (1)$$

$$\text{softmax}(f_c^{(s)}(\boldsymbol{x}_*)) = \frac{\exp(f_c^{(s)}(\boldsymbol{x}_*))}{\sum_c \exp(f_c^{(s)}(\boldsymbol{x}_*))}, \quad (2)$$

where S is the total number of Monte Carlo samples. The predicted class will be the one with the highest average probability, i.e., $c_* = \arg\max_c P(y_* = c \mid \boldsymbol{x}_*)$. At the end of these steps, the desired uncertainty criterion is applied. The pattern with the highest value for the criterion in question will be the one chosen to be labeled. The criteria used in this work are detailed in the next section.

2.3 Sample Selection Strategies for Active Learning

In the context of AL, uncertainty criteria are metrics used to identify which data samples are most valuable to label and add to the training set. The idea behind this selection is that

Table 1. Comparison of Active Learning Selection Strategies.

Strategy	Advantages	Disadvantages
Uncertainty Selection	Maximizes training efficiency by focusing on uncertain samples.	May select boundary samples that are not representative.
Diversity Selection	Ensures broad data coverage, preventing overfitting.	More complex to implement; may not always select the most informative samples.
Representativeness Selection	Samples are representative of the overall dataset.	May neglect outlier or boundary samples, limiting learning.
Margin Selection	Focuses on challenging samples, refining decision boundaries.	May overlook informative samples outside the margin.
Class Expansion Selection	Very effective in finding informative, difficult samples.	Requires multiple models, increasing computational cost.
Error-Expectancy Selection	Directly improves performance by focusing on impactful samples.	Error estimation can be difficult, especially with multiple classes.

the model has the greatest uncertainty in its predictions for the chosen samples, which means that labeling them can provide the greatest improvement in model learning [Pereira-Santos *et al*., 2019]. To determine this uncertainty, it is necessary to calculate the predicted class for each sample.

Least Confident. This criterion is defined as the complement of the highest average probability among the classes. It is a straightforward strategy to implement; however, in datasets with significant variations and noise, it may not be sufficient to improve the model. This criterion is calculated by:

$$\text{lc}(\boldsymbol{x}_*) = 1 - P(y_* = c_* \mid \boldsymbol{x}_*). \tag{3}$$

It is noted that a higher value for $\text{lc}(\boldsymbol{x}_*)$ is associated with a lower confidence in the final prediction.

Best and Second Best (BSB) Also called margin criterion, it measures the uncertainty of an ML model considering the difference between the two highest predicted probabilities for each sample, which represents a vulnerability. It is particularly effective in identifying samples where the model has difficulty distinguishing between two or more classes, especially in situations of similar confidence across multiple classes. The BSB is less sensitive to imbalanced classes compared to other criteria, providing a more balanced measure of uncertainty [Joshi *et al*., 2009]. Mathematically, for an input $\boldsymbol{x}_*$, let $P(y = c_1 \mid \boldsymbol{x}_*)$ be the predicted probability of the most likely class c_1, and $P(y = c_2 \mid \boldsymbol{x}_*)$ the predicted probability of the second most likely class c_2, the BSB-based uncertainty measure is defined by

$$\Delta(\boldsymbol{x}_*) = P(y = c_1 \mid \boldsymbol{x}_*) - P(y = c_2 \mid \boldsymbol{x}_*). \tag{4}$$

Entropy. Quantifies the uncertainty or disorder associated with a probability distribution. Samples with high entropy indicate greater model uncertainty, as they have more balanced probability distributions between classes. This criterion is useful for identifying regions of the input space where the model has lower confidence in its predictions. Entropy captures model uncertainty more completely, especially in situations with unbalanced probability distributions, however it can be computationally more expensive to calculate, especially in models with many labels, being sensitive to class imbalance, where uncertainty can be overestimated. for minority classes [Joshi *et al*., 2009]. For discrete distributions, as is the case with classification tasks, the entropy $H(\boldsymbol{x}_*)$ is defined by

$$H(\boldsymbol{x}_*) = -\sum_c P(y = c \mid \boldsymbol{x}_*) \log P(y = c \mid \boldsymbol{x}_*). \tag{5}$$

GPLCB - Gaussian Process Lower Confidence Bound. Estimates uncertainty by considering the mean of the predictive probabilities and the associated standard deviation. The deviation is estimated from the softmax values calculated from the Monte Carlo samples, being given by $\sigma_c(\boldsymbol{x}_*) = \sqrt{\mathbb{V}[\text{softmax}(f_c(\boldsymbol{x}_*))]}$, where $\mathbb{V}[\cdot]$ is the sample variance estimator, calculated considering the S samples $f_c^{(s)}(\boldsymbol{x}_*)$. Samples with a lower confidence limit indicate greater [Garnett, 2023] model uncertainty. The lower confidence limit is given by

$$\text{GPLCB}(\boldsymbol{x}_*) = 1 - (P(y = c_* \mid \boldsymbol{x}_*) - \beta\sigma_c(\boldsymbol{x}_*)), \tag{6}$$

where β is a parameter that regulates the size of the confidence interval to be considered.

3 Related Work

This section presents a literature review on ML and AL techniques applied to the VM process (detection and assessment phases), with a primary focus on the task of risk classification of vulnerabilities.

Elbaz *et al*. [2021] adopted the AL technique to label a dataset containing information about Common Platform Enu-

meration (CPE) identifiers, extracted from vulnerability descriptions published by NIST. The labeled data was used to train a ML model capable of classifying vulnerabilities into three distinct levels: (I) LOG, where the vulnerability is recorded without immediate alerts; (II) TICKET, where a ticket is generated for resolution during business hours; and (III) ALERT, indicating a critical vulnerability requiring immediate action. However, this solution has limitations, as it relies solely on CPE information, which is often unavailable at the time of vulnerability publication. Furthermore, it does not consider other crucial information about vulnerabilities, such as threat intelligence and specific context, which can impact the accuracy of alert decisions for security analysts.

Kure *et al*. [2022] proposed an integrated method for risk assessment in Cyber-Physical Systems (CPS) organized into three distinct stages. In the first stage, assets are classified using fuzzy logic to determine their criticality. Analysts answer five questions regarding the potential impact on the asset's Confidentiality, Integrity, and Availability (CIA), as well as the time required for recovery after an attack. In the second stage, a Machine Learning (ML) model is employed to predict the vulnerability of assets to ten specific types of cyberattacks. Finally, in the third stage, analysts assess the organization's security controls' compliance with the requirements established by ISO 27005 [Firoiu, 2015]. The experiments demonstrated the effectiveness of the asset criticality classification and the evaluation of security controls.

Kashyap *et al*. [2022] discussed the detection of cyberattacks in automotive traffic systems. The authors developed a model based on GP to identify malicious vehicles in mixed traffic environments. They also explored the possibility of investigating and integrating other anomaly detection methods and ML techniques to complement the GP-based model, aiming to enhance the ability to identify and respond to cyberattacks more broadly and effectively.

Sun *et al*. [2023] developed a framework called ASSBert, which combines AL and semi-supervised learning to detect vulnerabilities in smart contracts. The framework addresses the challenge of limited labeled data by using ML to efficiently select valuable data, improving the performance of the detection model. In experiments, the application outperformed conventional methods, even with a small amount of labeled data and a large amount of unlabeled data.

da Ponte *et al*. [2023] presented an AL-based methodology to develop a supervised model capable of emulating the expertise of specialists in vulnerability risk assessment. The study emphasized the importance of incorporating vulnerability information, threat intelligence, and contextual factors for effective risk evaluation, in contrast to inadequate practices that underestimate the likelihood and impact of vulnerability exploitation. The experiments demonstrated that the solution achieved high accuracy in identifying critical vulnerabilities, with performance comparable to human analysts, with only a slight difference of about 1%. The AL-based approach proved highly effective in risk classification, quickly outperforming random instance selection, even in scenarios with a large number of vulnerabilities. Additionally, it highlighted how the AL strategy facilitated more informed and efficient decision-making in comparison to random selection, ensuring a more precise and adaptive risk prioritization process.

A Framework for Risk Assessment, Prioritization, and Explainability of Vulnerabilities (FRAPE) proposes a framework for Risk-Based Vulnerability Management (RBVM), aiming to enhance the analysis, classification, and prioritization of vulnerabilities in organizations through AL and ML techniques [Ponte *et al*., 2025]. The framework consists of four main modules: data collection, vulnerability labeling with the help of AL, classification and prioritization using ML, and result interpretation. The FRAPE approach seeks to reduce complexity and human error by assisting security analysts in identifying the most critical vulnerabilities, allowing for effective prioritization for mitigation. The use of AL in FRAPE improves data labeling, while supervised learning emulates the expertise of analysts, making the risk management process more efficient.

While previous studies have advanced vulnerability detection and classification, they often focus on single techniques without exploring combinations of models or strategies, limiting the adaptability and robustness of the models. For example, approaches relying solely on methods like random forests or gradient boosting fail to leverage complementary models or selection strategies to improve performance.

The portfolio-based approach in this study is particularly effective in overcoming these challenges. By integrating diverse selection strategies, the model adapts to different stages of the learning process, increasing its robustness. For instance, when faced with limited labeled data or varying vulnerability characteristics, the model can balance exploration of uncertain areas and optimization of known regions using techniques like entropy, BSB, and GPLCB.

Moreover, the portfolio-based AL strategy addresses scalability issues inherent in traditional methods, where reliance on manual labeling or fixed strategies can be slow and inaccurate, particularly in large datasets. This approach allows the model to scale efficiently, reducing labeling effort and improving vulnerability prioritization for remediation. Its flexibility enables continuous adaptation to new information, providing a significant advantage in the ever-evolving cybersecurity landscape.

In this context, this work differs from others by employing advanced strategies for uncertainty estimation, combining AL with the non-parametric GP model. The use of GP is motivated by its ability to provide probabilistic predictions, essential for quantifying uncertainty in ML models. A key contribution of this study is the evaluation of various selection strategies, both individually and in combination, which has been a limitation in prior research. Table 2 shows the differences between these works and our approach. Experimenting with multiple strategies provides a more comprehensive understanding of the trade-offs between exploration and optimization, thereby enhancing decision-making.

4 Proposal

This study proposes an integrated solution using AL and GP for the risk classification of security vulnerabilities. The focus is to optimize data labeling, reducing the number of examples required for training and minimizing human effort. However, using a single selection strategy may be inadequate,

Table 2. Comparison between related works and the present study.

Reference	Context	Classifiers	Uncertainty Measurement	Portfolio
[Elbaz *et al*., 2021]	Risk Classification	CRFs	Least Confident	No
[Kure *et al*., 2022]	Risk Classification	KNN, NN, DT, RF, LR, NB	-	No
[Kashyap *et al*., 2022]	Vulnerability Detection	GPR	-	No
[Sun *et al*., 2023]	Vulnerability Detection	BERT, BERT-AL, BERT-SSL, ASSBERT	Entropy	No
[da Ponte *et al*., 2023]	Risk Classification	RF, GB, RL, SVC, MLP	Entropy	No
This Work	Risk Classification	GP	Entropy, Least Confident, BSB, GPLCB	Yes

CRFs: *Conditional Random Fields*, KNN: *K-Nearest Neighbors*, NN: *Neural Networks*, DT: *Decision Trees*, RF: *Random Forest*, LR: *Logistic Regression*, NB: *Naive Bayes*, BERT: *Bidirectional Encoder Representations from Transformers*, BERT-AL: *BERT with Active Learning*, BERT-SSL: *BERT with Semi-Supervised Learning*, ASSBERT: *An adversarially trained version of BERT*, RF: *Random Forest*, GB: *Gradient Boosting*, RL: *Reinforcement Learning*, SVC: *Support Vector Classification*, MLP: *Multilayer Perceptron*

since different regions of the data space may require different approaches.

To overcome this limitation, we propose the use of a portfolio-based approach, which combines multiple selection strategies, such as uncertainty selection, balancing exploration and exploitation. This allows for a more diverse coverage of samples, reducing bias and ensuring a more robust representation of the data.

The integration of GP with AL and the use of the portfolio approach offers an efficient solution, allowing the model to learn from a reduced number of samples and quickly adapt to new threats. This strategy optimizes human resources by focusing efforts on the most relevant samples, while reducing manual labeling effort. The following sections are dedicated to defining how the sample selection was performed considering the portfolio.

4.1 Portfolio-Based Approach

The portfolio-based strategy in AL uses multiple sample selection criteria during training, combining different approaches to ensure a balanced exploration of the data space. This allows the model to explore regions of high uncertainty while leveraging the knowledge already acquired, optimizing learning and focusing on the most informative instances for labeling.

By integrating several selection strategies (defined in section 2.3), the portfolio provides flexibility to deal with different uncertainty scenarios, ensuring a representative coverage of the data space and reducing the risk of bias in sample selection. This approach increases the learning efficiency, contributing to the generalization of the model.

In addition, the use of a portfolio optimizes the use of labeled data, reducing the need for large volumes of samples annotated by experts. By capturing different perspectives on model uncertainty, the portfolio improves overall performance, focusing on samples that cover unknown areas and less represented classes.

In the context of vulnerability risk classification, the application of a portfolio is innovative. Rather than relying on a single uncertainty metric, the model selects samples based on multiple uncertainty dimensions, which is crucial given the multiple factors that affect vulnerability criticality. The integration of multiple strategies ensures more efficient learning from the most relevant data, minimizing the labeling effort.

This research is one of the first to apply the portfolio approach in the cybersecurity domain. This application offers a novel solution to address the scarcity of labeled data, making the learning process more adaptable to the complexities of the vulnerability context.

4.2 Iterative Cycle of Active Learning with Portfolio of Strategies

The AL iterative cycle is a continuous and dynamic process, composed of several phases that repeat at each iteration. Figure 1 illustrates these steps, which are essential to improve the model's performance with a reduced number of labeled samples. The cycle is divided into the following phases:

(1) Initial Data: At the beginning of the cycle, we have two data sets: the labeled data set, which contains samples that are already known and labeled, and the unlabeled data set, composed of samples that have not yet been classified. The

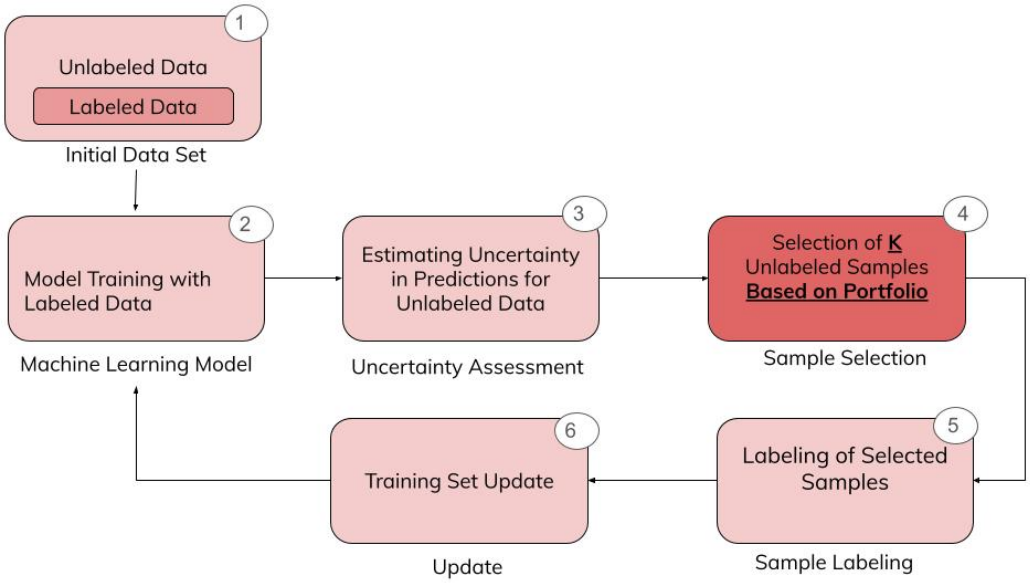

Figure 1. Iterative Cycle of Active Learning.

initial data set can be selected randomly or based on specific criteria (in this case, a study on the distribution of the data that will be used is feasible)

(2) Training the Machine Learning Model: With the labeled data set available, the ML model is trained. During this phase, the model adjusts its predictions to minimize discrepancies between the predicted classes and the actual labels of the training samples. This phase is crucial, as it creates the initial learning base that will guide the selection of the most informative samples.

(3) Estimating Uncertainty in Predictions for Unlabeled Data: After initial training, the model is applied to the unlabeled data set. Here, the model evaluates the uncertainty associated with each unlabeled sample. Uncertainty measures how confused the model is about the predictions for these samples. Instead of randomly labeling samples, the model focuses on the most uncertain ones, as these are the ones that, when labeled, can significantly improve its performance.

(4) Portfolio-Based Sample Selection: Once the uncertainty of the samples has been assessed, the portfolio of strategies comes into play. Instead of relying on a single selection strategy, the model can use a combination of selection strategies. As mentioned earlier, this allows for a more adaptive choice, balancing the exploration of uncertain areas and the optimization of well-understood areas. The model selects the K most informative samples according to these different uncertainty metrics.

(5) Labeling the Selected Samples: The K selected samples are then labeled. This process can be done by an expert who assigns the correct labels based on technical knowledge. In the experimental context, an oracle can be used to label the samples based on knowledge external to the model. This step is crucial because the labeled samples will be used to train the model in the next iteration.

(6) Updating the Training Set: After labeling, the newly labeled samples are added to the model's training set. This new dataset now includes both the previously labeled samples and the new samples. The model then performs a new training cycle using the updated data. This allows the model to refine its predictions by learning from the new labeled samples.

4.3 Sample Selection with Portfolio Strategies in Active Learning

Introducing a portfolio of strategies into AL significantly improves the dynamics of model training, especially in the selection of samples for labeling. While traditional methods rely on a single selection approach, combining multiple strategies allows for a more balanced exploration of the data space. This ensures that many regions of high uncertainty over areas with well-understood patterns are effectively explored.

This multi-strategy approach increases learning efficiency by prioritizing the most informative instances and avoiding overfitting to specific patterns or regions. By considering a combination of uncertainty metrics, the model explores diverse and underrepresented areas, quickly biases, and promotes a more comprehensive learning process.

A crucial aspect of this methodology is to avoid duplicate sample selection. The algorithm ensures that only unique and informative samples are chosen in each iteration, maximizing the efficiency of specialized labeling.

To implement this approach effectively, we implement the following algorithm to select k samples per strategy. It ensures that each strategy contributes independently to the training process, avoiding redundancies in selection. The central idea is to evaluate the informativeness of unlabeled instances using a predefined set of strategies, ensuring that each iteration enriches the labeled dataset with meaningful points.

This algorithm follows a structured approach to iterative sample selection, ensuring that each strategy independently selects the most relevant data. By maintaining a separate set of selected instances (L), duplication is avoided, ensuring that the sample set remains diverse and representative of all strategies.

Algorithm 1 Sample Selection

Require: Uncertainty estimates U for unlabeled data, dataset D, selection strategies S, number of samples per iteration k
Ensure: Selected instances stored in L
$L \leftarrow \emptyset$ ▷ Set of selected instances
$strategy\_samples \leftarrow \emptyset$
$strategy\_labels \leftarrow \emptyset$
for each $s \in S$ **do** ▷ Iterate over selection strategies
Compute selection scores Q using strategy s over U
Rank instances based on Q in descending order
$new\_samples \leftarrow \emptyset$
$new\_labels \leftarrow \emptyset$
for each instance i in ranked list **do**
if $i \notin L$ **then** ▷ Ensure no duplicates
Add i to $new\_samples$
Retrieve label for i from D and store in new_labels
end if
if size($new\_samples$) $= k$ **then**
break ▷ Stop after selecting k samples
end if
end for
$L \leftarrow L \cup new\_samples$
$strategy\_samples[s] \leftarrow new\_samples$
$strategy\_labels[s] \leftarrow new\_labels$
end for

This process improves the adaptability of the model throughout training. In the first few iterations, the exploration of the dataset is expanded, capturing diverse patterns. As learning progresses, the algorithm refines the selection based on uncertainty measures, optimizing the model's decisions and focusing on the most challenging regions.

In addition, this approach promotes computational efficiency by minimizing redundant computations. Each strategy selects exactly k samples per iteration, balancing the exploration of diverse areas of the data with the exploration of regions of greatest uncertainty. This balance ensures that the AL process remains robust and effective across different datasets and classification challenges.

By combining multiple strategies, the portfolio approach transforms the model's learning trajectory. This increases adaptability, improves efficiency, and strengthens generalization capabilities, resulting in more robust and accurate classifiers.

5 Experiments

The AL process iterates by selecting the most informative data instances for labeling. This iterative nature allows the model to progressively refine its predictions, minimizing the need for large amounts of labeled data. AL is iterative in that it requires feedback from domain experts during each iteration to label the most critical samples, improving overall performance even when data is limited or incomplete [Swiler *et al.*, 2020].

The proposed methodology uses a GP model within an AL framework to improve the vulnerability risk classification process. Random selection will be used as a baseline to evaluate the performance of sample selection strategies in active learning. Random selection involves choosing data samples without considering model uncertainty, providing a reference point for comparison. This establishes a performance benchmark, allowing the evaluation of more sophisticated methods that incorporate uncertainty quantification, such as those explored in AL.

The following sections define the dataset used, the established scenarios related to the model and the evaluation process.

5.1 Dataset

The CVEjoin dataset[2] was developed to assist cybersecurity analysts and researchers in the risk assessment and classification of vulnerabilities, enhancing decision-making processes related to vulnerability management. Unlike traditional datasets that use isolated information or metrics, such as CVSS metric, CVEjoin incorporates additional contextual threat intelligence, helping analysts understand the risk of vulnerability exploitation more effectively.

The dataset is built by aggregating data from several trusted sources, including the NVD, ExploitDB, MITRE's CWE, and others. The initial dataset includes over 200,000 vulnerabilities, with information to reflect the vulnerabilities published between 2002 and 2022. This information is gathered through web scraping, API queries, and downloading JSON files from various trusted sources. Automated tools, written in Python, utilize libraries such as urllib [3] and BeautifulSoup [4] for data mining.

The dataset contains 208 labeled samples, each with 29 attributes and categorized into four risk classes: Low, Moderate, Important, and Critical. This classification is based on the risk posed by the vulnerability, determined by both its intrinsic characteristics (e.g., CVSS score, impact on CIA) and external factors like exploitability (e.g., public exploits, attack vector).

These risk labels are vital for prioritizing vulnerability remediation and are intended to help organizations decide where to allocate resources effectively in the face of potential cyberattacks.

Additionally, the dataset underwent a pre-processing phase to manage its diverse attributes, which include continuous, discrete, boolean, and categorical values. Categorical data were converted to numerical values using the *one-hot encoding* technique, as there is no inherent hierarchy among these categories. Where necessary, the data was also normalized to have zero mean and unit variance, ensuring consistency and facilitating model training.

The CVEjoin dataset plays a pivotal role in our approach, enabling the classification of vulnerabilities based on their risk of exploitation. This classification allows us to streamline critical processes, such as prioritizing vulnerability remediation, optimizing patch management strategies, and efficiently

[2] CVEJoin Security Dataset, `https://github.com/rodrigoparente/cvejoin-security-dataset`
[3] URLIB, `https://docs.python.org/3/library/urllib.html`
[4] Beautiful Soup Documentation, `https://beautiful-soup-4.readthedocs.io/en/latest/`

allocating cybersecurity resources. Furthermore, the dataset is used to evaluate the performance of our GP model combined with AL and PA strategies. This approach enhances the model's accuracy, adaptability, and efficiency in real-world cybersecurity scenarios.

5.2 Machine Learning model evaluation

To evaluate the performance of the AL strategies and the ML model, several key metrics will be used.

Accuracy measures the proportion of correct predictions (true positives and true negatives) out of the total number of samples. While useful, it can be misleading in imbalanced datasets. Precision quantifies the model's ability to avoid false positives, calculated as the proportion of true positives among predicted positives. It answers, "How many of the predicted positives were actually correct?" Recall assesses the model's ability to identify all true positives, calculated as the proportion of true positives among all actual positives. It is crucial in cases where missing positives (false negatives) is costly, such as in cybersecurity. F1-Score combines precision and recall into a single metric, providing a balanced measure that penalizes extreme values of either metric. It is particularly useful in imbalanced datasets. Finally, the AUC (Area Under the ROC Curve) evaluates how well the model discriminates between classes. A higher AUC indicates better class distinction, with a value closer to 1.0 reflecting a more effective model.

5.3 Experimental Setup and Model Configurations

To carry out this study, the GPyTorch [5] library was used, which optimizes the use of hardware for GP model applications. Table 3 describes the configurations used for AL. **Initial Size** represents the amount of labeled data used to train the model before starting the AL process, which is scaled proportionally to the number of classes. **Active Iterations** represent the number of AL cycles executed, where new samples are selected and added to the training set. **Active Selection per Iteration** configures the number of samples chosen in each iteration to be labeled and included in training. **Selection Strategies** represent the different methods for selecting the most informative samples (described in section 2.3). Experiments were carried out to evaluate AL strategies repeatedly independently, increasing the statistical robustness and reliability of the results. With more repetitions, the mean and standard deviation of performance metrics provide a more accurate estimate of the model's actual performance, reducing random bias caused by specific splits of the data.

Likewise, Table 4 presents the configurations used by GP. **Number of Tasks** refers to the number of simultaneous tasks that the GP model needs to solve. In this case, it is double the number of classes to allow for multi-task modeling. **Learning Rate** is a parameter that controls the speed of model adjustment during training in order to guarantee model convergence. **Number of Induction Points** is used to approximate GP on large data sets. **Likelihood Samples** represent the number of samples used in both training and testing to estimate the likelihood of the data, influencing the accuracy and robustness of the model. **Number of Epochs** represents the total number of complete passes through the training data set. **Kernel** is used to measure the similarity between data, with the RBF kernel, given by:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp(-\frac{1}{2}\sum_d w_d^2(x_d - x'_d)^2) \qquad (7)$$

a common choice in the literature. The values σ_f^2 and $w_d^2|_{d=1}^D$, where D is the input dimension, are hyperparameters automatically adjusted during the training itself [Rasmussen and Williams, 2006; Hensman *et al*., 2013] .

Table 3. Active Learning settings used in the experiments.

Settings	Value
Initial Size	$10\times$ Number of classes
Active Iterations	Scenario-dependent
Active Selection by Iteration	Scenario-dependent
Selection Strategies	Least Confident, Entropy, BSB, GPLCB
Independent repetitions	30
Data split (Train/Test)	90%/10%

Table 4. Gaussian Process model configurations.

Configurations	Value
Number of tasks	$2 \times$ number of classes
Learning rate	0.05
Number of inducing points	$5 \times$ number of classes
Monte Carlo samples (train)	100
Monte Carlo samples (test)	1000
Number of epochs	3000
Kernel	RBF

The configurations used in the experiments were carefully selected to provide a fair and balanced assessment of the model's performance and active learning strategies. However, it is important to note that these configurations resulted in increased execution times, as each repetition involved running a full set of active learning experiments. This process includes retraining the model, selecting samples for labeling, and evaluating performance, all of which require substantial computational resources.

In terms of the AL approach, it will be shown in the following section that better performance is related to scenarios where a small number of data points are labeled at a time. This study explored several AL configurations simulating different levels of intervention by an information security expert in the classification of vulnerability risk. The impact of the expert's interaction frequency and the number of iterations on the accuracy of vulnerability risk classification was evaluated through three distinct scenarios.

The first scenario, Scenario I, involves 100 iterations, each requiring the labeling of a single vulnerability. While this scenario provides a detailed and progressive analysis, it demands significant time and effort from both the expert and the system, as the model needs to be retrained after each iteration.

[5] GPyTorch's documentation, `https://docs.gpytorch.ai/en/stable/`

In Scenario II, the number of iterations is reduced to 20, with each iteration involving the labeling of 5 vulnerabilities. This setup aims to balance expert interaction with the efficiency of the AL process, requiring less time from the expert while still maintaining reasonable accuracy.

Finally, Scenario III minimizes expert involvement by reducing the number of iterations to 10, each involving the labeling of 10 vulnerabilities. Although this scenario reduces the frequency of model retraining, it may impact the overall accuracy due to the lower frequency of expert interaction. Table 5 summarizes the configuration of the analyzed scenarios.

Table 5. Active Learning Scenarios: Iterations and Selection Settings

Scenario	Active Iterations	Active Selection
Scenario I	100	1
Scenario II	20	5
Scenario III	10	10

It is important to note that the random selection strategy is used as a baseline for comparing the performance of other AL strategies. Since random selection disregards model uncertainty and relevance, it is expected to yield less optimal results than more advanced methods that focus on selecting the most informative samples.

6 Evaluation of Individual Sample Selection Strategies

This section presents tables with the mean and standard deviation of performance metrics for different sample selection strategies at the end of the AL cycle. It also includes graphs comparing mean accuracy curves across iterations of model selection, labeling, and retraining. Additionally, the AUC metric is calculated to quantify performance throughout iterations, unlike other metrics that only reflect the final state. All metrics were computed based on 30 independent repetitions to ensure statistical robustness.

6.1 Scenario I Results

Table 6 presents the results of the mean and standard deviation of the metrics achieved by the GP classification model at the end of the cycle of different AL strategies in scenario I. In this scenario, there are 100 AL iterations, where in each iteration, one vulnerability is selected at a time by the strategy and labeled by an expert.

When analyzing the results, it is observed that the random strategy (random selection) presents the worst performance with the lowest overall values. On the other hand, it is noteworthy that all other AL strategies that use some uncertainty calculation present better metrics than Random, and are close to each other. The differences in performance between the strategies can be attributed to the way each of them selects the examples for labeling, with random selection not providing these benefits, which is reflected in lower values in the evaluation metrics.

Table 6. Mean and standard deviation of different evaluation metrics for the GP classification model considering various active learning strategies in Scenario I.

Strategy	Accuracy ($\mu \pm \sigma$)	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F1-score ($\mu \pm \sigma$)
BSB	0.78 ± 0.05	0.83 ± 0.06	0.78 ± 0.05	0.78 ± 0.06
Entropy	0.77 ± 0.05	0.86 ± 0.06	0.77 ± 0.05	0.77 ± 0.06
GPLCB	0.76 ± 0.05	0.80 ± 0.02	0.76 ± 0.06	0.76 ± 0.07
LC	0.78 ± 0.04	0.82 ± 0.06	0.77 ± 0.04	0.78 ± 0.05
Random	0.75 ± 0.05	0.78 ± 0.06	0.75 ± 0.05	0.74 ± 0.06

Considering the four metrics (accuracy, precision, recall, and F1-score) together, it is evident that the BSB strategy performs the best overall. While BSB and Entropy have similar average values across these metrics, the AUC curve played a crucial role in the selection of BSB as the top-performing strategy. The AUC for BSB (0.682) was slightly higher than that of Entropy (0.679), indicating a better ability to discriminate between classes throughout the AL process. This additional metric highlights that BSB not only identifies true positives effectively but also maintains better class separation compared to Entropy, especially in cases where class boundaries are unclear. Furthermore, BSB demonstrated a more balanced measure of uncertainty, being less sensitive to class imbalances than other strategies.

Figure 2 shows the average accuracy curves of the best performing strategy (BSB) and the *baseline* strategy (Random), as a function of the number of labeled data with their respective AUCs. With this graph, one can observe the behavior of the strategies throughout the entire AL iterative cycle.

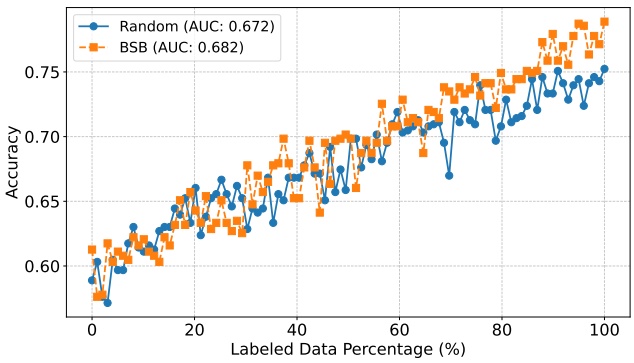

Figure 2. Average accuracy and AUC as a function of the number of labeled data in Scenario I.

As expected, it is observed that the accuracy of the vulnerability risk classification model increases as new samples selected by the AL strategies are labeled by the expert and used to retrain the model. Furthermore, it is observed that as this process progresses with more labeled data, the BSB strategy begins to outperform Random, culminating in a final accuracy of 0.78, as observed in table 6. The greater accuracy of BSB throughout the AL iterations is quantitatively demonstrated with a higher AUC value when compared to the Random strategy, as reported in the graph legend.

6.2 Scenario II Results

Table 7 presents the results of scenario II, where 20 AL iterations are performed and 5 vulnerabilities are labeled at a time by the expert. As observed in the previous scenario, the selected selection strategy is also inferior to the other strategies, and BSB presents the best overall analysis.

Table 7. Mean and standard deviation of different evaluation metrics for the GP classification model considering various active learning strategies in Scenario II.

Strategy	Accuracy ($\mu \pm \sigma$)	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F1-score ($\mu \pm \sigma$)
BSB	0.77 ± 0.05	0.73 ± 0.06	0.77 ± 0.05	0.76 ± 0.06
Entropy	0.76 ± 0.05	0.72 ± 0.06	0.76 ± 0.05	0.75 ± 0.06
GPLCB	0.75 ± 0.05	0.74 ± 0.02	0.75 ± 0.06	0.75 ± 0.07
LC	0.75 ± 0.04	0.72 ± 0.06	0.75 ± 0.04	0.75 ± 0.05
Random	0.74 ± 0.05	0.72 ± 0.06	0.74 ± 0.05	0.74 ± 0.06

Figure 3 displays the graph with the average accuracy curves and the respective AUCs of the BSB and Random strategies as a function of the number of labeled data during the iterative process. Once again, it is noted that BSB surpasses Random in terms of average accuracy in the final AL iterations, reaching an accuracy of 0.77 and an AUC of 0.652.

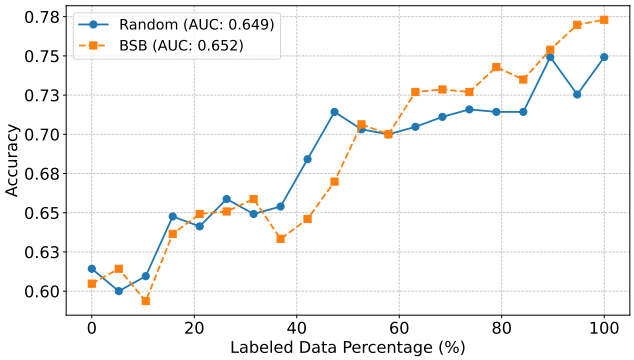

Figure 3. Average accuracy and AUC as a function of the number of labeled data in Scenario II.

6.3 Scenario III Results

Table 8 and Figure 4 display the results of scenario III, where fewer AL iterations are performed (ten) and more vulnerabilities are labeled at a time by the expert (ten). A similar result to the other two scenarios is observed: the BSB strategy obtained the best performance indicators in the vulnerability risk classification, while Random presented the worst performance. However, there is a more prominent gain in the performance of BSB compared to Random in the average accuracy and AUC curves throughout the AL process, as can be seen in Figure 4.

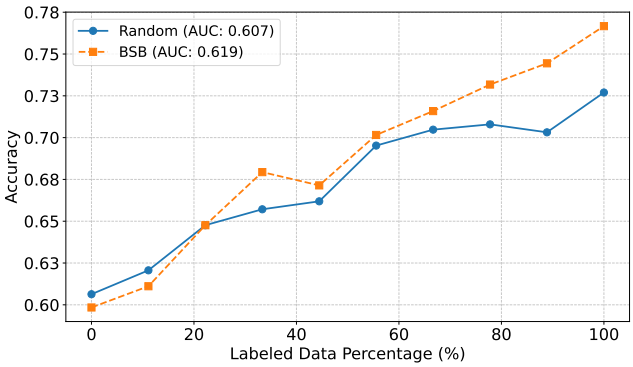

Figure 4. Average accuracy and AUC as a function of the number of labeled data in Scenario III.

Table 8. Mean and standard deviation of different evaluation metrics for the GP classification model considering various active learning strategies in Scenario III.

Strategy	Accuracy ($\mu \pm \sigma$)	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F1-score ($\mu \pm \sigma$)
BSB	0.76 ± 0.05	0.74 ± 0.07	0.76 ± 0.05	0.76 ± 0.06
Entropy	0.74 ± 0.05	0.74 ± 0.07	0.74 ± 0.05	0.74 ± 0.06
GPLCB	0.74 ± 0.05	0.76 ± 0.07	0.74 ± 0.05	0.73 ± 0.06
LC	0.76 ± 0.05	0.74 ± 0.07	0.76 ± 0.05	0.75 ± 0.04
Random	0.72 ± 0.04	0.71 ± 0.07	0.72 ± 0.04	0.72 ± 0.04

6.4 Best strategy for selecting samples individually

After conducting experiments in the three scenarios, the BSB strategy proved to be the most effective for vulnerability risk classification, outperforming the other techniques in most performance metrics, particularly in Scenarios II and III. Therefore, this subsection focuses on the detailed analysis of this strategy during the labeling and retraining process within the active learning framework.

Figure 5 shows the average accuracy curves and AUCs for the BSB strategy as a function of the number of labeled data points across all the previous scenarios. Scenario I appeared to be the most robust and specific, characterized by labeling one sample at a time, which requires more processing time but provides the best evaluations and the highest AUC. However, it is worth noting that the accuracy curves are quite similar, which leads us to conclude that by increasing the number of labels and reducing the time for labeling, we still obtain good results while significantly reducing processing time and computational cost.

7 Evaluation of Portfolio-Based Sample Selection Strategy

The strategy portfolio approach allows for the simultaneous selection of samples based on different criteria, promoting diversity and reducing bias in the selection of labeled data. However, this approach is only applicable in Scenarios II and III, where the number of samples selected per iteration is aligned with the experiment setup.

In Scenario II, the total number of selections per iteration is

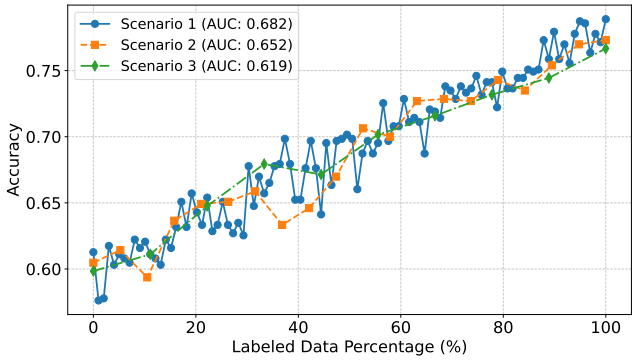

Figure 5. Average accuracy and AUC of the BNB strategy according to the number of labeled data for the different scenarios analyzed.

5. Given that the portfolio consists of 5 strategies (BSB, Random, Entropy, GPLCB, and Least Confident), each selects one sample ($k = 1$), ensuring an exact total of 5 samples per iteration, which matches the expected setup. In Scenario III, where the expected total number of selections per iteration is 10, the same five strategies select two samples each ($k = 2$), resulting in the required 10 samples per iteration.

In Scenario I, the portfolio approach cannot be applied, since only one sample can be selected per iteration. If each strategy selected one sample, the total would be 5 samples per iteration, exceeding the predefined limit of the experiment. Restricting the selection to a single strategy per iteration would contradict the concept of balanced and simultaneous selection among strategies, making the portfolio approach infeasible in this scenario.

Therefore, the strategy portfolio approach is feasible only in Scenarios II and III, where k can be adjusted to maintain consistency in the number of selections. In Scenario I, the restriction of one sample per iteration prevents the configuration of a portfolio.

Table 9 summarizes the selection strategy for each scenario, illustrating the constraints and corresponding values of k that ensure alignment with the experimental setup.

Table 9. Portfolio selection strategy across different scenarios.

Scenario	Total selection	Samples per strategy (k)
I	1	Not applicable
II	5	1
III	10	2

The following subsections will show the main results found in this approach. It will be possible to notice that there was greater consistency in accuracy values.

7.1 Scenario II Results

In Scenario II, the portfolio-based strategy emerged as the most effective method, outperforming others across all evaluation metrics. This strategy stands out due to its ability to balance exploration and exploitation, ensuring that the most informative samples are selected at each iteration, thereby contributing to more efficient learning. By combining multiple selection criteria, the portfolio strategy avoided overfitting to specific regions of the data, leading to higher accuracy and AUC compared to other strategies.

While the BSB strategy also delivered strong results, it primarily focused on regions of high uncertainty, limiting the diversity of selected samples. On the other hand, the portfolio strategy's combination of different selection criteria provided a more balanced approach, reducing bias and enhancing generalization. As shown in Table 10, the portfolio strategy outperformed both BSB and Random in terms of accuracy, recall, and f1-score.

Table 10. Mean and standard deviation of different evaluation metrics for the GP classification model considering the Portfolio (K=1), BSB, and Random on Scenario II

Strategy	Accuracy ($\mu \pm \sigma$)	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F1-score ($\mu \pm \sigma$)
BSB	0.77 ± 0.05	0.73 ± 0.06	0.77 ± 0.05	0.76 ± 0.06
Portfolio	0.79 ± 0.08	0.74 ± 0.11	0.77 ± 0.08	0.77 ± 0.09
Random	0.74 ± 0.05	0.72 ± 0.06	0.74 ± 0.05	0.74 ± 0.06

As a baseline, the Random strategy showed the poorest performance, underscoring the fact that purely random selection does not effectively contribute to model training. This highlights the importance of incorporating uncertainty-driven strategies, which focus on areas with the highest uncertainty for more efficient data labeling.

Furthermore, the results demonstrated that the portfolio strategy's diverse sample selection improved the decision boundaries. Figure 6 presents the average accuracy and AUC over the active learning iterations, illustrating that the portfolio strategy consistently maintained the highest performance, especially compared to the Random strategy.

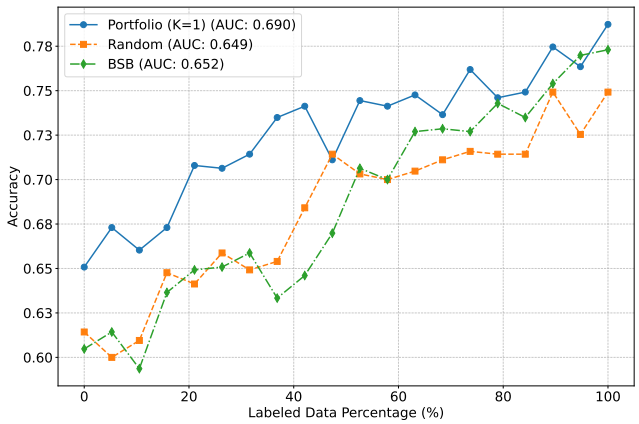

Figure 6. Average Accuracy and AUC as a function of the number of labeled data in Scenario II for K = 1.

A key advantage of the portfolio-based strategy is its ability to prioritize both informative and representative samples, allowing the model to refine its decision-making in areas of high uncertainty. Figure 7 illustrates the average number of selected samples per strategy, showing that Class 2 was the most frequently selected by uncertainty-driven strategies (Entropy, GPLCB, and Least Confident). This prioritization occurs because Class 2 exhibited high ambiguity, making it a target for strategies focused on reducing uncertainty.

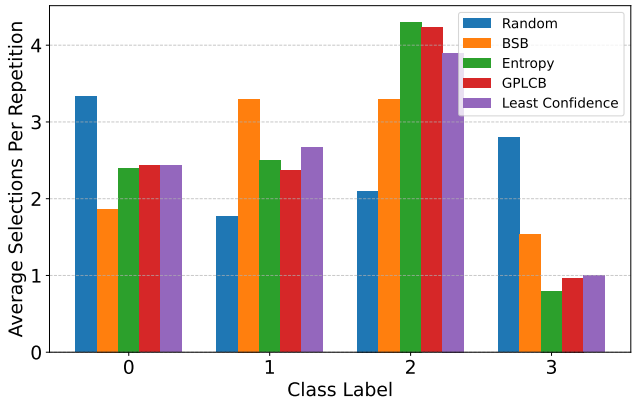

Figure 7. Average Class Selections per Strategy (average over 30 repetitions) for Portifolio in Scenario II (k=1)"

Uncertainty-based strategies like Entropy and GPLCB focused heavily on Class 2, while BSB took a more balanced approach, selecting both Class 1 and Class 2 samples nearly equally. This strategy balanced the representation across all classes, reducing bias and improving the model's generalization ability. The Random strategy, as expected, maintained a more uniform selection across all classes, serving as a baseline for comparison.

Additionally, the confusion matrix in Figure 8 supports the effectiveness of the portfolio strategy. The highest classification accuracy was observed for Classes 0 and 3, which were easier for the model to classify. These classes were selected less frequently by uncertainty-based strategies since the model already showed high confidence in classifying them.

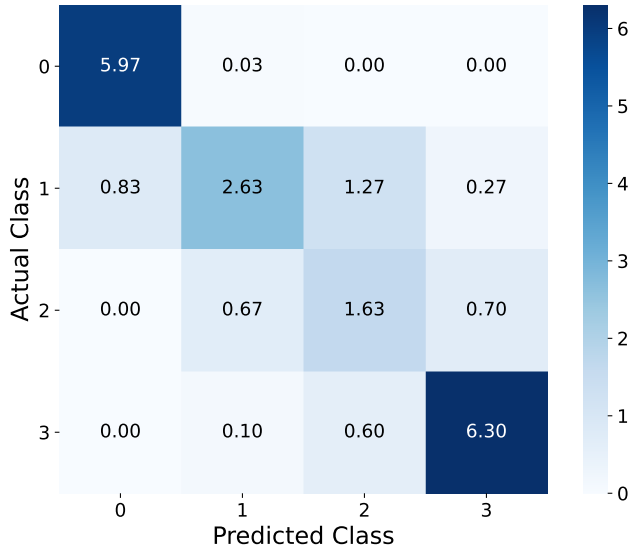

Figure 8. Confusion Matrix (average over 30 repetitions) for Portifolio in Scenario II (K=1)

Class 2 showed the highest confusion rate, frequently being misclassified as Class 1 or Class 3. This confirms the necessity of prioritizing Class 2 with uncertainty-driven strategies, which focus on improving the decision boundary where the model faces the greatest challenge.

The analysis of the confusion matrix further validates the strategic effectiveness of the portfolio. By focusing on classes where uncertainty was highest, the portfolio ensured that the most critical data points were labeled, optimizing the efficiency of active learning. This led to balanced model improvement, reducing errors in difficult classes while reinforcing decision boundaries.

7.2 Scenario III Results

In Scenario III, where K=2, the active learning process allows each strategy to select two samples per iteration. This increases the volume of labeled data over time, enabling more refined data acquisition. The results show that the portfolio strategy balanced exploration and exploitation, effectively optimizing the model's learning curve.

Figure 9 illustrates the evolution of average accuracy and AUC (Area Under the Curve) values for each strategy as a function of the percentage of labeled data. The portfolio strategy (K=2) achieved the highest AUC (0.66), outperforming both the Random strategy (AUC: 0.607) and the BSB strategy (AUC: 0.619). These findings suggest that the combined selection of multiple strategies contributes to more efficient learning, helping the model converge to better performance with fewer labeled samples.

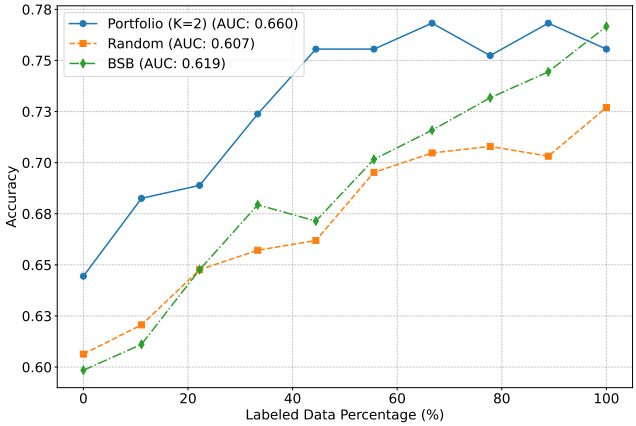

Figure 9. Average Accuracy and AUC as a function of the number of labeled data in Scenario III for K = 2.

Table 11 summarizes the mean and standard deviation of the evaluation metrics for the BSB, Random, and Portfolio strategies. As shown, the Portfolio strategy outperformed the individual strategies across all metrics. These results highlight the advantage of combining multiple selection criteria, which not only account for uncertainty but also ensure representativeness and diversity in the selected samples.

Table 11. Mean and standard deviation of different evaluation metrics for the GP classification model considering the Portfolio (K=2), BSB, and Random strategies on Scenario III.

Strategy	Accuracy ($\mu \pm \sigma$)	Precision ($\mu \pm \sigma$)	Recall ($\mu \pm \sigma$)	F1-score ($\mu \pm \sigma$)
BSB	0.76 ± 0.05	0.74 ± 0.07	0.76 ± 0.05	0.76 ± 0.06
Portfolio	0.79 ± 0.07	0.76 ± 0.12	0.79 ± 0.07	0.77 ± 0.09
Random	0.72 ± 0.04	0.71 ± 0.07	0.72 ± 0.04	0.72 ± 0.04

The higher values obtained by the Portfolio strategy sug-

gest that combining multiple selection criteria allows for a more balanced distribution of labeled samples across the feature space. This approach helps refine the model's decision boundaries more effectively compared to BSB and Random, which primarily focus on uncertainty-driven sample selection.

The portfolio strategy's improvement in recall (0.79) indicates that the model was better at correctly identifying positive instances, showing a significant gain in generalization capacity. These results underscore the value of combining uncertainty-based strategies with others that ensure representativeness and diversity in the learning process.

Figure 11 shows the average class selections per strategy, revealing that Class 2 was the most frequently selected by uncertainty-based strategies (Entropy, GPLCB, and Least Confident). The prioritization of Class 2 stems from the high uncertainty surrounding this class, which makes it a focal point for active learning strategies aimed at reducing uncertainty.

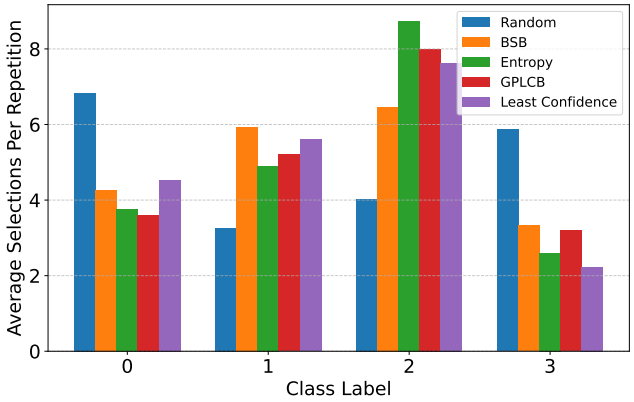

Figure 10. Average Class Selections per Strategy (average over 30 repetitions) for Portifolio in Scenario III (k=2)

Uncertainty-based strategies, such as Entropy and GPLCB, heavily prioritized Class 2, while the BSB strategy took a more balanced approach, selecting both Class 1 and Class 2 samples almost equally. This approach ensured that the model was not biased towards a single class, promoting a more generalizable model. Random sampling, on the other hand, exhibited a more uniform selection pattern across all classes, which serves as a baseline for comparison.

The confusion matrix in Figure 10 illustrates the classification performance of the model. Classes 0 and 3 achieved the highest classification accuracy, showing that these categories were easier for the model to learn. These classes were selected less frequently by uncertainty-based strategies, as the model already exhibited high confidence in classifying them.

Class 2, on the other hand, showed the highest confusion rate, with frequent misclassifications as Class 1 or Class 3. This result reinforces the necessity of prioritizing Class 2 with uncertainty-driven strategies. These strategies focused on improving the decision boundaries where the model had difficulty making accurate predictions.

Class 1 exhibited moderate performance, with a confusion rate influenced by its similarity to Class 2, further justifying the need for BSB's balanced selection approach. This prevented Class 1 from being neglected during active learning.

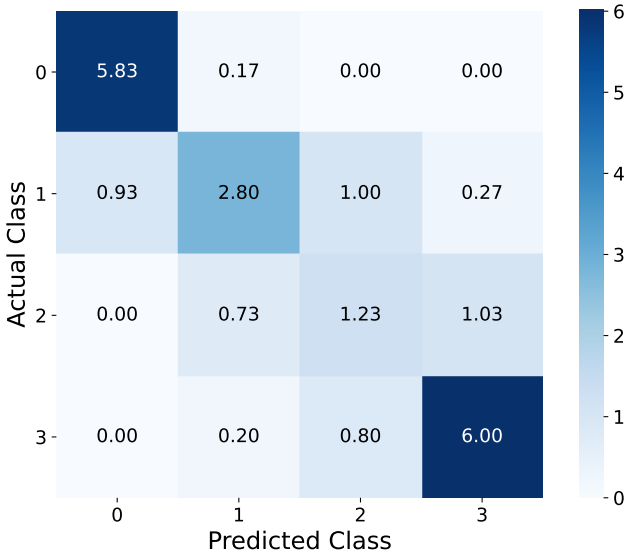

Figure 11. Confusion Matrix (average over 30 repetitions) for Portifolio in Scenario III (K=2)

The analysis of selection patterns and the confusion matrix reaffirms that the portfolio strategy efficiently directs data acquisition to the most challenging regions of the decision space. Uncertainty-based strategies focused on refining the decision boundaries for Class 2, while BSB ensured representative selection across the dataset. The presence of fewer selections for Classes 0 and 3 highlights the model's confidence in these categories, optimizing the learning process and preventing resources from being spent on already well-understood data points.

7.3 Portfolio Strategy Performance in Different Scenarios

The comparison between Scenarios II and III further highlights the effect of increasing the number of selections per iteration. In Scenario II (K=1), where only one sample was selected per iteration, the model's accuracy grew more slowly as data acquisition was more gradual. The uncertainty strategies focused heavily on Class 2, which the model struggled to classify. The confusion matrix confirmed that misclassifications between Classes 1 and 2 were a major challenge.

In Scenario III (K=2), where two samples were selected per iteration, model learning was accelerated, leading to faster improvements in performance. The portfolio strategy's ability to select more diverse samples contributed to this enhanced performance, as evidenced by the reduced uncertainty in the confusion matrix.

Ultimately, the findings suggest that increasing the number of selections per iteration enhances active learning efficiency. However, the success of this approach is contingent on the quality of the selected samples, highlighting the importance of the portfolio strategy in diversifying and balancing sample selection. This methodology maximized knowledge acquisition, resulting in better model performance and more precise decision boundaries.

Figure 12 shows a comparison of the performance of the six different strategies in terms of F1-score. Observing the image,

it is possible to notice that the BSB and Portfolio strategies stand out with the highest medians, indicating a superior average performance compared to the others. In addition, the Portfolio box is relatively narrow, suggesting less variability in the results, that is, greater consistency. On the other hand, the Random strategy has the lowest median and a wider box, which reflects a lower and less stable performance.

The presence of outliers, such as isolated points above the boxes in some strategies (for example, Least Confidence), indicates that, although most of the results are concentrated in a specific range, there are exceptional cases with significantly better performance. This visualization allows us to conclude that strategies such as BSB and Portfolio are the most recommended for achieving high and consistent F1-scores, while the Random strategy, as expected, presents the worst results. The analysis reinforces the importance of choosing active methods instead of random ones for tasks that demand precision and reliability.

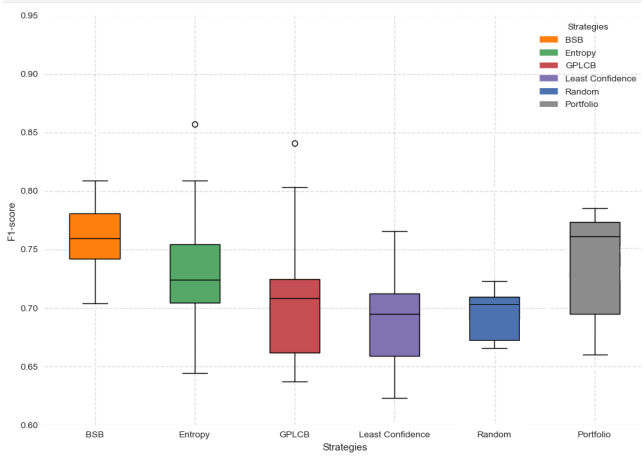

Figure 12. F1-Score in Scenario III (K=2)

8 Discussion and Limitations

This section presents some limitations related to this work. Although the Gaussian Process-based active learning model presents significant advantages for vulnerability risk classification, zero-day scenarios present some important limitations that should be considered.

Zero-day vulnerabilities, by definition, lack information at the time of discovery, making it difficult for probabilistic models to accurately infer risk. Incomplete or poorly representative data can compromise the effectiveness of predictions. Therefore, the ability of the model to generalize to new vulnerabilities depends on the diversity and quality of the training dataset. When faced with new combinations of attributes or previously unobserved contexts, the risk assessment may be uncertain or inaccurate. This highlights the importance of comprehensive and up-to-date data, including information on impacts and early exploits.

To maintain relevance and accuracy, the model should be constantly fed with newly discovered vulnerabilities and their actual behavior. Without this continuous cycle of updating, it may become less effective at identifying patterns in emerging threats. Active learning is only effective if there is a dynamic flow of information for progressive adjustment. In addition to the technical and contextual attributes captured by the model, the probability of exploitation can depend on unpredictable external factors, such as human actions or new attack techniques. This limits its ability to predict risks with complete reliability, especially when dealing with new threats. Finally, vulnerabilities with completely new characteristics can generate initial periods of high uncertainty, until the model accumulates enough information to provide more reliable assessments. Adaptability directly depends on the speed and scope of updates to the dataset, ensuring that the system remains effective in the face of constant changes in the cybersecurity landscape.

In real-world scenarios, our machine learning model with active learning is better suited to Vulnerability Management systems than Security Information and Event Management (SIEM). While SIEMs require real-time analysis of large volumes of dynamic data—which can limit the effectiveness of complex or human-dependent models—VM operates in a more analytical and periodic context. Here, the model can prioritize vulnerabilities based on criteria such as criticality and likelihood of exploitation, refining its predictions through continuous feedback (active learning), without the pressure of immediate responses.

In addition, VM deals with more structured data (such as vulnerability scans and CVE databases), where false positives are less critical and there is room for manual validation. Active learning excels in this environment because it allows the model to "learn" from experts over time, improving its accuracy in risk classification. In SIEMs, the need for speed and adaptation to emerging threats makes the model less viable, reinforcing that its natural application is to support proactive decision-making in VM.

9 Concluding Remarks

This research explored the applicability of a Gaussian Process-based Machine Learning model combined with Active Learning and the use of the Porftolio Strategy for classifying security vulnerabilities in information technology systems based on their exploitation risk.

The methodology was evaluated across three experimental scenarios, each varying in the number of vulnerabilities labeled per AL iteration. The findings demonstrated that AL strategies effectively reduce the human effort required for vulnerability labeling without compromising the accuracy of risk classification models. A comparative analysis of AL strategies revealed that sample selection based on model uncertainty is more effective than random selection, emphasizing the importance of incorporating uncertainty to optimize the learning process.

The experiments also highlighted that when using the portfolio approach, the number of samples selected per iteration significantly impacts the model's learning efficiency. In scenarios where only one sample was labeled per iteration (K = 1), the model's improvement was more gradual, requiring more iterations to achieve competitive performance. In contrast, when two samples per iteration were selected (K = 2),

the model's learning process accelerated, achieving higher accuracy with fewer iterations. This shows that grouping samples into larger labeling rounds improves learning efficiency while reducing computational overhead and expert interaction, making the AL process more scalable.

Furthermore, the study concluded that minimizing the number of AL iterations while grouping the same data volume into larger batch labels optimizes the efficiency of model retraining and minimizes human intervention in data labeling. Future research will focus on extending the ML-based vulnerability risk classification framework to other domains such as web applications and cloud environments, as well as further exploring the portfolio-based sample selection process.

Declarations

Authors' Contributions

Davyson S. Ribeiro: *Methodology:* Contributed to the methodology and statistical analysis. *Writing – Original Draft:* Wrote the first draft of the manuscript. *Writing – Review and Editing:* Contributed to the revisions and improvements of the manuscript. *Related Work:* Conducted research on related works to identify gaps that could be addressed with the approach presented.

Rafael S. Lemos: *Methodology:* Assisted in designing the experimental setup and data analysis. *Investigation:* Conducted the experiments and data collection. *Writing – Review and Editing:* Reviewed and edited sections of the manuscript.

Francisco R. P. da Ponte: *Investigation:* Conducted related work and collected data. *Data Curation:* Organized and prepared data for analysis. *Writing – Review and Editing:* Contributed to manuscript revisions.

César Lincoln C. Mattos: *Conceptualization:* Led the conception and design of the research. *Methodology:* Contributed to the development of the experimental approach and evaluation of the machine learning model. *Writing – Original Draft:* Helped write parts of the manuscript. *Writing – Review and Editing:* Reviewed and provided suggestions for improvement. Also contributed to the revision of the techniques applied in the manuscript.

Emanuel B. Rodrigues: *Conceptualization:* Provided key insights into the research design. *Project Administration:* Supervised the project and coordinated the research team. *Writing – Review and Editing:* Contributed to the revision of the techniques applied in the manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets (and/or softwares) generated and/or analysed during the current study are available in `https://github.com/rodrigoparente/cvejoin-security-dataset`.

References

Alshaya, F. A., Alqahtani, S. S., and Alsamel, Y. A. (2023). Vrt: A cwe-based vulnerability report tagger: Machine learning driven cybersecurity tool for vulnerability classification. In *2023 IEEE/ACM 1st International Workshop on Software Vulnerability (SVM)*, pages 10–13. IEEE. DOI: 10.1109/svm59160.2023.00007.

Blasco, T., Sánchez, J. S., and García, V. (2024). A survey on uncertainty quantification in deep learning for financial time series prediction. *Neurocomputing*, 576:127339. DOI: 10.1016/j.neucom.2024.127339.

da Ponte, F. R., Rodrigues, E. B., and Mattos, C. L. (2023). A vulnerability risk assessment methodology using active learning. In *International Conference on Advanced Information Networking and Applications*, pages 171–182. Springer. DOI: 10.1007/978-3-031-28451-9$_1$5.

Elbaz, C., Rilling, L., and Morin, C. (2021). Automated risk analysis of a vulnerability disclosure using active learning. In *C&ESAR 2021-28th Computer & Electronics Security Application Rendezvous*, pages 1–19. Available at:`https://ceur-ws.org/Vol-3056/paper-04.pdf`.

Firoiu, M. (2015). General considerations on risk management and information system security assessment according to iso/iec 27005: 2011 and iso 31000: 2009 standards. *Quality-Access to Success*, 16(149). Book.

Foreman, P. (2019). *Vulnerability management*. Auerbach Publications. DOI: 10.1201/9780429289651.

Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press. DOI: 10.1017/9781108348973.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013*. AUAI Press. DOI: 10.48550/arxiv.1309.6835.

Hensman, J., Matthews, A., and Ghahramani, Z. (2015). Scalable variational gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR. DOI: 10.48550/arxiv.1411.2005.

Hoffman, M., Brochu, E., De Freitas, N., *et al*. (2011). Portfolio allocation for bayesian optimization. In *UAI*, pages 327–336. DOI: 10.48550/arXiv.1009.5419.

Hore, S., Shah, A., and Bastian, N. D. (2023). Deep vulman: A deep reinforcement learning-enabled cyber vulnerability management framework. *Expert Systems with Applications*, 221:119734. DOI: 10.1016/j.eswa.2023.119734.

Jakkal, V. (2022). Cybersecurity threats are always changing—staying on top of them is vital, getting ahead of them is paramount. Available at:`https://www.microsoft.com/en-us/security/blog/2022/02/09/cybersecurity-threats-are-always-changing-staying-on-top-of-them-is-vital-getting-ahead-of-them-is-paramount/` Microsoft Security Blog.

Joshi, A. J., Porikli, F., and Papanikolopoulos, N. (2009). Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pages 2372–2379. IEEE. DOI: 10.1109/cvpr.2009.5206627.

Kashyap, A., Chakravarthy, A., and Menon, P. P. (2022). Detection of cyber-attacks in automotive traffic using macroscopic models and gaussian processes. *IEEE Control Systems Letters*, 6:1688–1693. DOI: 10.1109/lcsys.2021.3131259.

Kure, H. I., Islam, S., Ghazanfar, M., Raza, A., and Pasha, M. (2022). Asset criticality and risk prediction for an effective

cybersecurity risk management of cyber-physical system. *Neural Computing and Applications*, 34(1):493–514. DOI: 10.1007/s00521-021-06400-0.

Pereira-Santos, D., Prudêncio, R. B. C., and de Carvalho, A. C. (2019). Empirical investigation of active learning strategies. *Neurocomputing*, 326:15–27. DOI: 10.1016/j.neucom.2017.05.105.

Ponte, F. R. P., Rodrigues, E. B., and Mattos, C. L. C. (2025). Frape: A framework for risk assessment, prioritization and explainability of vulnerabilities in cybersecurity. *Journal of Information Security and Applications*, 89:103971. DOI: 10.1016/j.jisa.2025.103971.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning.* Adaptive computation and machine learning. MIT Press. DOI: 10.7551/mitpress/3206.001.0001.

Ross, R. S. (2012). Guide for conducting risk assessments. Special Publication 800-30 Rev. 1, National Institute of Standards and Technology.

Sabottke, C., Suciu, O., and Dumitras, T. (2015). Vulnerability disclosure in the age of social media: Exploiting twitter for predicting {Real-World} exploits. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 1041–1056. Available at:`https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/sabottke`.

Sun, X., Tu, L., Zhang, J., Cai, J., Li, B., and Wang, Y. (2023). Assbert: Active and semi-supervised bert for smart contract vulnerability detection. *Journal of Information Security and Applications*, 73:103423. DOI: 10.1016/j.jisa.2023.103423.

Swiler, L. P., Gulian, M., Frankel, A. L., Safta, C., and Jakeman, J. D. (2020). A survey of constrained gaussian process regression: Approaches and implementation challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2). DOI: 10.1615/jmachlearnmodelcomput.2020035155.

Tenable (2023). Três desafios reais enfrentados pelas organizações de segurança cibernética. Available at:`https://www.tenable.com`.

Vasconcelos, T. d. P., de Souza, D. A. R. M. A., Mattos, C. L. C., and Gomes, J. P. P. (2019). No-past-bo: Normalized portfolio allocation strategy for bayesian optimization. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 561–568. DOI: 10.1109/ICTAI.2019.00084.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA. DOI: 10.7551/mitpress/3206.001.0001.