International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Enhancing Cloud-Based Smart Contract Security: A Hybrid AI and Optimization Approach for Vulnerability Prediction in FinTech

Ranadeep Reddy Palle[1], Haritha Yennapusa[2], Krishna Chaitanya Rao Kathala[3]

[1]Independent Researcher, Software Engineer, MS in Computer Science
[2]MS in Computer Science
[3]PhD in Computer Science

Abstract: ***Financial industries operate within a framework of strict regulatory requirements, making compliance a top priority. Smart contracts, integral to the operations of FinTech companies, must align with these regulations. Cloud-based platform offers security as a service (SecaaS) to the scalable and cost-effective solution for analyzing, monitoring, and predicting vulnerabilities in smart contracts. This approach allows FinTech firms to concentrate on their core services while benefiting from specialized security tools. The potential consequences of smart contract vulnerabilities, such as financial losses, fraud, or data manipulation, underscore the critical need for proactive prediction and mitigation. By addressing vulnerabilities in advance, FinTech platforms can prevent financial losses and uphold the integrity of their transactions. Given that FinTech platforms handle customer funds, sensitive financial information, and automated transactions, maintaining trust and reliability is paramount. Predicting vulnerabilities plays a pivotal role in building and sustaining trust among users and stakeholders. This study introduces a hybrid artificial intelligence and optimization technique for smart contract vulnerability prediction in FinTech. The modified barnacles mating optimization (MBMO) algorithm is employed for the extraction of complex syntactic and semantic features, enhancing the accuracy of vulnerability predictions. Additionally, the general regressive artificial neural network (GR-ANN) is utilized to predict vulnerabilities, specifically describing vulnerability types in smart contracts deployed in a cloud environment. The evaluation of this framework involves rigorous testing using the ScrawID-real Ethereum smart contract benchmark dataset, demonstrating its capability and accuracy in predicting smart contract vulnerabilities. The study introduces a novel hybrid artificial intelligence and optimization technique aimed at predicting vulnerabilities in cloud-based smart contracts, specifically in the FinTech sector. Utilizing the modified barnacles mating optimization algorithm and the general regressive artificial neural network, this approach enhances the accuracy of vulnerability detection. The paper demonstrates the methods efficacy through rigorous testing with the ScrawID-real Ethereum smart contract benchmark dataset, highlighting its potential to bolster security in FinTech applications.***

Keywords: smart contract, vulnerability prediction, cloud computing, artificial intelligence, machine learning, optimization technique

1. Introduction

In the realm of cloud storage services, providers furnish clients with on-demand storage services, available in the form of Infrastructure as a Service (IaaS) or Software as a Service (SaaS) [1]. Numerous research endeavors have been directed towards enhancing the efficiency, reliability, and user-friendliness of cloud storage services.. The cloud storage industry has recently burgeoned into a significant market, with predictions indicating growth from $23.48 billion in 2016 to an estimated $88.91 billion by 2020 [2]. The persistent rise in Internet of Things (IoT) devices, the diversification of computation-intensive services and the escalating demand for enterprise mobility continue to fuel the popularity of cloud storage [3]. Despite its manifold advantages, concerns about the security, reliability, and privacy of cloud storage persist as serious issues. Cloud service providers typically construct storage data centers as distributed systems employing commodity hardware, making them susceptible to both independent and correlated failures [4]. Despite employing various hardware and software techniques to prevent data corruption and ensure security and reliability, occasional data corruption incidents still occur. In such cases, clients seek assurance that their data is securely stored, reliable, and unaltered on the cloud. Traditional integrity assurance methods [5], such as hash functions and signatures, face limitations in cloud storage scenarios as they often require clients to possess full copies of the data. Therefore, it becomes imperative to devise specific methods for verifying data reliability and integrity in cloud storage scenarios. Cloud audit protocols, particularly provable data possession schemes, have been introduced to address this need [6].

In the current optimization operational framework of the integrated energy market [7], the involvement of energy managers, load aggregators, and similar third-party entities act as intermediaries in successfully executing energy trading and ensuring system stability. These entities often leverage demand-side response and game theory methods for energy trading and scheduling, aiming to achieve optimal operation of the integrated energy system and garner benefits [8]. However, the gains of these intermediaries fundamentally originate from energy suppliers and users, and their optimal scheduling strategies may lack complete credibility. Some intermediaries may subjectively categorize users' loads as translatable, reducible, or convertible during demand response [9]. Still, the actual integrated energy system faces challenges in accurately quantifying these load types, making it challenging to fully adhere to the instructions provided by intermediaries. Simultaneously, the existing integrated energy market falls short of adequately supporting the growing carbon trading market, especially given the limited number of physical carbon exchanges [10]. There is an urgent need to establish a trustworthy and independent cloud service platform to replace the intermediary function and cater to the demands of the carbon trading sector.

Our contributions

In our research, we present a hybrid artificial intelligence and optimization technique designed for predicting smart contract vulnerabilities in the FinTech domain. The primary contributions of our proposed approach are outlined below.

1) We leverage the MBMO algorithm to extract intricate syntactic and semantic features. This algorithm plays a crucial role in enhancing the accuracy of vulnerability predictions in smart contracts. By focusing on both syntactic and semantic aspects, MBMO contributes to a more comprehensive understanding of potential vulnerabilities.
2) The GR-ANN is incorporated into our framework to predict vulnerabilities, specifically providing detailed descriptions of vulnerability types within smart contracts deployed in a cloud environment. This neural network model is trained to recognize and categorize various vulnerability patterns, contributing to a more nuanced and precise prediction process.
3) To assess the effectiveness of our framework, we conduct thorough testing utilizing the ScrawID-real Ethereum smart contract benchmark dataset. This dataset serves as a representative sample of real-world scenarios, allowing us to validate the capability and accuracy of our approach in predicting smart contract vulnerabilities. Through this evaluation, we aim to demonstrate the practical applicability and reliability of our proposed technique in the FinTech context.

The rest of this paper is organized as follows. Section 2 describes the review of recent works related to smart contract vulnerability detection. Section 3 presents the detailed description and working process of the proposed framework. In addition, Section 4 illustrates the results and comparative analysis of smart contract vulnerability detection methods. Finally, the paper concludes in Section 5.

2. Related works

2.1 State-of-art works

Xu et al. [11] have introduced a machine learning-based analytical model for identifying smart contract vulnerabilities. Their approach involves the utilization of shared child nodes to enhance the analysis. They employed abstract-syntax-trees (ASTs) for smart contracts, and vulnerabilities from two datasets, namely SmartBugs and SolidiFI-benchmark. This feature vector formed the basis of their machine-learning model. It's worth noting that this method necessitates expertise in smart contracts, encompassing knowledge of their syntax and semantics. The results were impressive with accuracy for their K-nearest neighbors (KNN) model all exceeding 90%.

Liu et al. [12] have introduced a model for detecting vulnerabilities in smart contracts that combines hybrid deep learning with classical expert patterns in a transparent and interpretable manner. This system integrates neural networks with tools designed for the automatic extraction of expert patterns. Their approach is rooted in established expert rules, although it's important to note that manually defined patterns come with the inherent risk of potential errors. The results demonstrated substantial improvements in accuracy, enhancing it from 84% to 90% when it comes to detecting vulnerabilities, surpassing the performance of state-of-the-art models.

Zhang et al. [13] have introduced a method for predicting vulnerabilities in smart contracts, using an ensemble learning (EL) approach that incorporates seven different neural networks. This model focuses on detecting vulnerabilities at the contract level using data related to contract vulnerabilities. This method was evaluated using a target dataset created from the IG, and its performance was compared with that of static tools and seven independent data-driven methods. The verification and comparative results indicate that the SCVDIE method outperforms other data-driven methods in terms of accuracy and robustness when it comes to predicting smart contract vulnerabilities in the specified task.

Qian et al. [14] have introduced an approach that merges the bidirectional long short-term memory (Bi-LSTM) with an attention mechanism for the simultaneous detection of multiple vulnerabilities in smart contract op-codes. They initially preprocessed the data to transform op-codes into a feature matrix suitable for input into the neural network. Subsequently, they applied the Bi-LSTM model integrated with an attention mechanism to categorize smart contracts based on multiple labels. The experimental outcomes show the model's capability to detect multiple vulnerabilities simultaneously, with all evaluation metrics surpassing the 85% mark.

Griggs et al. [15] have proposed blockchain-based smart contracts to work with secure examination and the leading group of clinical sensors. Using a private blockchain based on the Ethereum protocol, we developed a system in which the sensors communicate with a smart device that calls smart contracts and writes records of all events to the blockchain. By sending notices to patients and medical professionals and keeping a safe record of who has initiated these activities, this brilliant agreement framework would maintain ongoing patient monitoring and clinical mediations.

Huang et al. [16] have created a model for detecting vulnerabilities in smart contracts, using a multi-task learning approach. The bottom sharing layer primarily focuses on learning the semantic information embedded within the input contract. The text representation is transformed into a vector through a combination of word and positional embedding techniques. A neural network, incorporating an attention mechanism, is applied to learn and extract the feature vector from the contract. The task-specific layer plays a central role in executing individual tasks.

Zhang et al. [17] have introduced a CBGRU model which combines various word embeddings with DL techniques, including LSTM, GRU, BiLSTM, CNN, and BiGRU. By employing these diverse DL models, the model effectively extracts features and combines them to detect vulnerabilities in smart contracts. SmartCheck plays a pivotal role in converting Solidity source code into an intermediate display format based on XML and subsequently assesses it against

XPath patterns. Fuzzy testing tools are also integrated into the vulnerability detection process for smart contracts.

Xing et al. [18] have focused on three shortcomings of smart contracts: has_short_address, has_flows and is_greedy. For the three kinds of shortcomings, they used a cutting matrix, one more technique to eliminate shortcoming components and construct three shortcoming area models for assessment. The preliminary outcomes show that the distinguishing proof precision considering mind association and slice structure is better than that taking into account cerebrum association and opcode features.

Oliva et al. [19] have driven an exploratory examination of clever arrangements. Remarkably as opposed to before assessments that focused on unambiguous pieces of a subset of wise arrangements are to have a greater understanding of all arrangements that are by and by conveyed in Ethereum. They see that the movement level is focused on a tiny number of agreements. Expressly, only 0.05% of the sagacious agreements are the goal of 80% of the trades that are delivered off arrangements.

Wang et al. [20] have proposed a high openness and bound together data channel-based secure design for Ethereum smart agreement. FSFC is planned to allow the sent splendid arrangements to continue to run routinely regardless of when faced with likely attacks. Before being handled, terrible sources of info progressively distinguished and wiped out utilizing a methodology. FSFC sent and survey utility using certifiable splendid concurrences with known number shortcomings.

2.2 Research gaps

Predicting vulnerabilities in smart contracts within the FinTech domain, especially in a cloud-based environment, is confronted with several challenges. The issue of data privacy and security arises, as cloud environments involve the handling of sensitive financial information. Entrusting third-party cloud providers with such confidential data introduces concerns about privacy and security. The integration of smart contract vulnerability prediction systems with regulatory compliance standards is crucial in the highly regulated FinTech environment. Ensuring alignment with financial regulations is imperative for building trust in and gaining acceptance of these systems. The dynamic and evolving nature of smart contracts poses another significant challenge. Smart contracts undergo frequent updates and modifications, necessitating continuous monitoring and adaptation of prediction models. Maintaining accuracy and explainability is the third challenge, given the financial risks involved in the FinTech sector. Interpretability of prediction models is vital for accountability and transparency, contributing to user trust. Resource utilization and scalability are also critical considerations in cloud-based systems. Efficient use of computational resources, along with the ability to scale the system to handle a large volume of smart contracts, is crucial for real-world deployment. Moreover, the dependency on external cloud providers introduces a level of vulnerability, as factors such as downtime or changes in provider policies can impact the availability and reliability of the prediction system. The FinTech industry's diversity in platforms and technologies raises the challenge of achieving interoperability and standardization in cloud-based smart contract vulnerability prediction. Seamless integration with different FinTech applications requires standardized approaches. Continuous monitoring and adaptation to the evolving landscape of smart contracts and emerging threats is another facet that demands attention. Finally, a holistic approach is needed to address these challenges, combining advanced technical solutions with a deep understanding of the regulatory and operational aspects of the FinTech industry.

3. Proposed Methodology

The purpose of this study is to develop and validate a hybrid artificial intelligence and optimization technique for predicting vulnerabilities in cloud-based smart contracts, with a focus on enhancing security in the FinTech domain. This research is significant as it addresses the critical need for advanced security measures in FinTech smart contracts by predicting vulnerabilities effectively contributes to the trust and reliability of financial transactions on cloud platforms.

3.1 Background study

Fig. 1 shows the system architecture designed for the implementation of a hybrid artificial intelligence and optimization technique tailored for smart contract vulnerability prediction in the FinTech domain. This architecture is meticulously structured into four essential steps, each playing a pivotal role in fortifying the security and dependability of smart contracts. The initial step, modal generation, serves as a foundational process crucial for comprehending and evaluating the structure and behavior of smart contracts. This phase entails the transformation of the source code of smart contracts into diverse modal representations. These modal representations act as distinct viewpoints of the smart contract's data and are derived directly from the source code. Specifically, the modal generation process takes the source code as input and produces two fundamental modalities: the operation code and the control flow diagram. The operation code encapsulates the fundamental instructions and actions inherent in the smart contract, while the control flow diagram provides a visual representation of the contract's logic and the sequence of operations it executes. Following modal generation, the subsequent step involves feature extraction. Here, the modified barnacles mating optimization (MBMO) algorithm comes into play, contributing to the extraction of intricate syntactic and semantic features. This enhancement significantly augments the accuracy of vulnerability predictions. Additionally, the general regressive artificial neural network (GR-ANN) is harnessed to predict vulnerabilities, offering a detailed description of vulnerability types specifically within smart contracts deployed in a cloud environment. The culmination of this systematic approach is the rigorous evaluation of the framework. This evaluation entails meticulous testing utilizing the ScrawID-real Ethereum smart contract benchmark dataset. The outcomes of these assessments not only showcase the framework's capability but also highlight its exceptional accuracy in predicting vulnerabilities within

smart contracts, thereby affirming its efficacy in the FinTech domain.

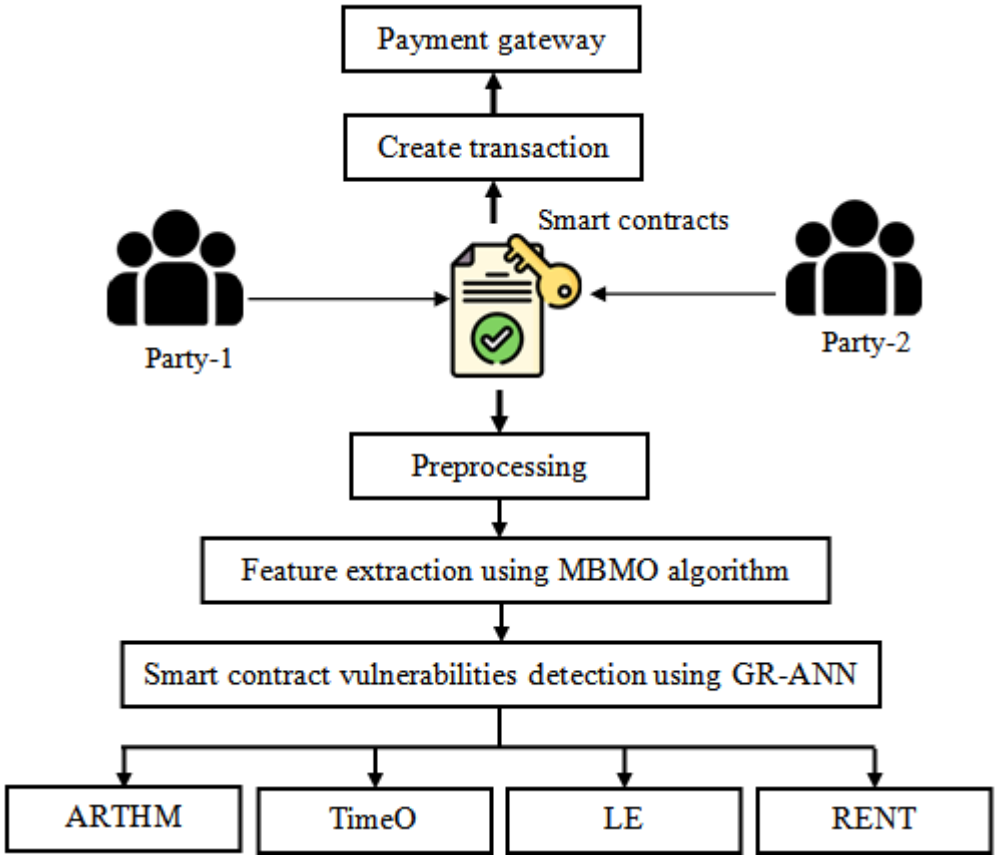

Figure 1: System model of proposed smart contract vulnerability prediction in FinTech

3.2 Feature extraction

Feature extraction is a pivotal stage in the process of predicting vulnerabilities in smart contracts, and the modified barnacles mating optimization (MBMO) algorithm plays a crucial role in enhancing the accuracy of this extraction. This algorithm is specifically designed to unravel complex syntactic and semantic features embedded within the source code of smart contracts. Syntactic features refer to the structural elements of the code, focusing on the arrangement and composition of the programming constructs. The MBMO algorithm, through its optimization techniques, efficiently identifies and extracts these syntactic features. This includes patterns, keywords, and other structural components that contribute to the overall syntax of the smart contract. Semantic features, on the other hand, delve into the meaning and functionality of the code. They encompass the logical relationships and operations performed by different parts of the code. The MBMO algorithm, being tailored for mating optimization, excels in deciphering these intricate semantic features. It identifies the underlying logic, conditional statements, and the overall behavior of the smart contract.

The objective function of the MBMO algorithm is to minimize the total power loss in the power system network while meeting all the set constraints. For loss minimization, the basis of the formulation can be described as follows:

$$f = \min Q_{loss}(y,u) = \sum_{l=1}^{nl} Q_{loss} \quad (1)$$

where Q_{loss} is the real power loss calculated in each transmission line in MW, Nl is the total transmission lines in the power system network, y and u are the vector of dependent variables and control variables to be optimized respectively. This expression is subject to the equality, $h(y,u)$ and inequality constraints, $g(y,u)$ that need to be fulfill and expressed as follow:

$$h(y,u) = 0 \quad (2)$$

$$g(y,u) \le 0 \quad (3)$$

The equality constraint is the power balanced of load flow which can be expressed

$$Q_{Hj} - Q_{dj} = V_j \sum_{i \in n_j} V_i (H_{ji} \cos\theta_{ji} + A_{ji} \sin\theta_{ji}) \quad (4)$$

$$P_{Hj} - P_{dj} = V_j \sum_{i \in n_j} V_i (A_{ji} \cos\theta_{ji} + H_{ji} \sin\theta_{ji}) \quad (5)$$

where A_{ji} is the susceptance of line j-i, H_{ji} is the conductance of line j-i, V_j is voltage at bus-j and V_i is the voltage at bus-i. Q_{Hj} and P_{Hj} on the other hand are the real and reactive power generation, Q_{dj} and P_{dj} are the real and reactive power demand respectively.

$$V_{Hj}^{Min} \le V_{Hj} \le V_{Hj}^{Max} \quad j = 1,...n_H \quad (6)$$

$$Q_{Hj}^{Min} \le Q_{Hj} \le Q_{Hj}^{Max} \quad j = 1,...n_H \quad (7)$$

$$P_{Hj}^{Min} \le P_{Hj} \le P_{Hj}^{Max} \quad j = 1,...n_H \tag{8}$$

where n_H is the maximum number of generators. Reactive compensation elements must be operated within the limits, as follows:

$$P_{Cj}^{Min} \le P_{Cj} \le P_{Cj}^{Max} \quad j = 1,...n_H \tag{9}$$

where n_C is the number of the reactive elements installed in the system. Transformer tap settings must be operated within the limits, as follows:

$$S_j^{Min} \le S_j \le S_j^{Max} \quad j = 1,...n_S \tag{10}$$

where n_S is the number of transformers. Barnacles are classified and recognized as sessile organisms which living deep in the ocean.

$$y_j^n -^{new} = qy_{barnacle\_m}^n + qy_{barnacle\_d}^n \quad for \quad K \le ql \tag{11}$$

$$y_j^n -^{new} = rand() + y_{barnacle\_m}^n \quad for \quad K > ql \tag{12}$$

where $K = |barnacle\_m - barncle\_d|$, q is the normally distributed pseudo random number, $p = (1-q)$, $y_{barnacle\_m}^n$ and $y_{barnacle\_d}^n$ are the randomly chosen variables for barnacle's parents respectively. Meanwhile, rand() denotes the random number range between zero to one (0~1). By referring to these equations, q and p represent the inheritance percentage from the respective barnacles' parents.

3.3 Detection and classification

Detection and classification of vulnerabilities in smart contracts are pivotal steps in ensuring the security of these digital agreements, especially when deployed in a cloud environment. In this context, the general regressive artificial neural network (GR-ANN) emerges as a powerful tool for predicting vulnerabilities and providing detailed descriptions of the identified vulnerability types. During the detection phase, the GR-ANN takes the enriched feature set obtained through MBMO-based extraction and processes it through its neural network architecture. The network's hidden layers work to uncover intricate patterns and correlations within the feature data. By leveraging regression techniques, the GR-ANN can assign a continuous value to predict the likelihood or severity of vulnerabilities associated with different parts of the smart contract. The output of the GR-ANN provides not only a prediction of vulnerability presence but also detailed descriptions of the identified vulnerability types. These descriptions can include specifics such as the nature of the vulnerability, potential risks, and recommended mitigation strategies. The ability of the GR-ANN to offer nuanced insights into the diverse landscape of vulnerabilities enriches the predictive capabilities of the overall system. GR-ANN consists of input, hidden, summation, and division layers. A frequently employed method for the analysis of complicated time series is the phase space reconstruction method, which is extracted from the embedding theorem presented. Suppose there is a time series a(T) and according to the embedding theorem, it can be expanding as follows:

$$s(T) = [a(T), a(T-c), a(T-2c), ...a(T-(Ep-1)c)] \tag{13}$$

The two important parameters in phase space reconstruction are Ep and c are calculated in this work using the correlation dimension and the mutual information.

$$s_k(T) - [a_k(T), a_T(T-c_k), a_k(s-2c_k), ...a_k(s-Ep_k-1)c_k)] \tag{14}$$

The details of employing both the correlation dimension method and mutual information method to get the appropriate values of Ep and c have been described above. The theory of kernel regression is the basis of GR-ANN. Suppose that the joint probability density function of V and U is $F(V_s, U)$. The regression of u on V_s is given as follows.

$$U(V_s) = e[U \mid V_s] = \frac{\int_{-\infty}^{\infty} UF(V_s, U)cU}{\int_{-\infty}^{\infty} F(V_s, U)cU} \tag{15}$$

When $F(V_s, U)$ is not recognized, it have to typically be calculated from a sample of V and b. By using Parzen distribution free, the function $F(V_s, U)$ can be found based on sample data collection {VI, UI} n i=1.

$$F(V_s, U) = \frac{1}{q(2\pi)^{\frac{h+1}{2}} z_1, z_2 ... z_h z_U} \sum_{I=1}^{q} E^{-C(V_s, V_I)} E^{-C(U, U_I)} \tag{16}$$

$$C(V_s, V_I) = \frac{(V_s, V_I)^2}{2\sigma^2} \tag{17}$$

$$C(U, U_I) = \frac{(U, U_I)^2}{2\sigma^2} \tag{18}$$

The critical parameter of the GR-ANN called the smooth parameter (σ) has a considerable influence on its ability to predict. σ establishes the function's broad, which characterizes the area of impact and, hence, the total examples are taken into account to evaluate a variable. For a large value of σ, further examples will be deemed. Solving the two integrals of produces the following:

$$U(V_s) = \frac{\sum_{I=1}^{q} U_I E^{-C(V_s, V_I)}}{\sum_{I=1}^{q} E^{-C(V_s, V_I)}} \tag{19}$$

So the forecasted value $U(V_s)$ is the weighted mean of all of the recorded values UI . The topology of the GR-ANN consists of 4 levels of the processing unit, as indicated.

$$a_I^k = aLy, I + s * (aUy, I - aLy, I) \tag{20}$$

which is the most important one, a new harmony (solution) is created from the stored harmonies in GR-ANN using any mixture of the three different rules.

$$a_I^{'} = \begin{cases} a_I^{'} \in \left\{a_I^1, a_I^2, ...a_I^{hms}\right\} & hmcr \geq s3 \\ a_I^{'} \in [aLy, I, a_{Uy}, I] & otherwise \end{cases} \quad (21)$$

Then, each harmony vector selected from GR-ANN is examined to decide whether it would be pitch-adjusted based on the optimal parameter as follows:

$$pitch\,adjusting\,rule\,for\,a_I^{'} = \begin{cases} + & par \geq s3 \\ - & otherwise \end{cases} \quad (22)$$

If the decision is yes for any $a_I^{'}$, the $a_I^{'}$ value is updated as follows:

$$a_I^{'} = a_I^{'} + s4 \times YZ \quad (23)$$

The new optimal solution is assessed in the previous step and GR-ANN is revised by replacing its worst solution with the new one. Lastly, the procedures are repeated until the stopping criterion is met.

4. Results and Discussion

In this section, we present a detailed results and comparative analysis of our novel cloud-based smart contract vulnerability prediction method designed for FinTech, along with an evaluation of existing methodologies. The assessment involved a meticulous series of tests conducted on a publicly accessible dataset to gauge the overall effectiveness of the proposed approach. Specifically, we employed the ScrawID dataset, a real Ethereum smart contract dataset annotated with vulnerability information. The evaluation focused on detecting four prevalent types of contract vulnerabilities: ARTHM, TimeO, LE, and RENT. To acquire the source code of the corresponding contracts, we utilized a web crawler to retrieve data from Ethscan, resulting in a comprehensive dataset comprising 9252 smart contracts. The entire implementation of our proposed method was carried out using Python and SIF tools within a system environment featuring Ubuntu 18.04, Python 3.10, Scikit-learn 1.2.2, and PyTorch 1.13.1. The experimental setup included a computer equipped with an Intel Xeon Gold 6240R CPU running at 2.6 GHz, a Tesla V100S-32 GB GPU, and 64 GB of RAM. In our comparative analysis, we juxtaposed the outcomes of our GR-ANN method against those of established methods, including Mythril [12], PSO-NDS [13], ASG-TL [14], ANN-CL [16], and SmartCheck [17]. This extensive evaluation provides a robust understanding of the efficacy of our methods in comparison to state-of-the-art frameworks, shows their strengths and contributions to the field.

Table 1: Comparative analysis of proposed and existing smart contract vulnerability prediction methods for "ARTHM-vulnerability"

Methods	Metrics (%)					
	Accuracy	Precision	Recall	Specificity	F-measure	AUC
Mythril [12]	79.293	77.753	77.854	77.575	77.803	77.975
PSO-NDS [13]	82.947	81.407	81.508	81.229	81.457	81.629
ASG-TL [14]	86.601	85.061	85.162	84.883	85.111	85.283
ANN-CL [16]	90.255	88.715	88.816	88.537	88.765	88.937
SmartCheck [17]	93.909	92.369	92.470	92.191	92.419	92.591
GR-ANN	97.563	96.023	96.124	95.845	96.073	96.245

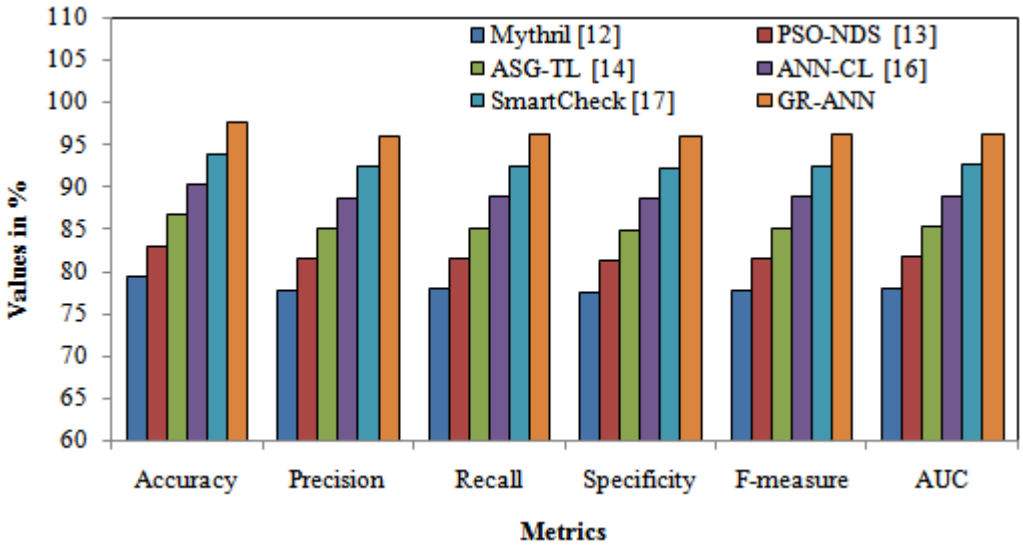

Figure 2: Results comparison for ARTHM-vulnerability prediction

The comparative analysis of smart contract vulnerability prediction methods, specifically targeting the "ARTHM-vulnerability," is presented in Table 1. Each method's performance is evaluated across various metrics, providing a comprehensive overview of their accuracy, precision, recall, specificity, F-measure, and AUC. Starting with the Mythril method, it demonstrates an accuracy of 79.293%, precision of 77.753%, recall of 77.854%, specificity of 77.575%, F-measure of 77.803%, and AUC of 77.975%. Moving to the PSO-NDS method, there is a noticeable improvement across

all metrics, with an accuracy of 82.947%, precision of 81.407%, recall of 81.508%, specificity of 81.229%, F-measure of 81.457%, and AUC of 81.629%. ASG-TL exhibits further enhancements, achieving an accuracy of 86.601%, precision of 85.061%, recall of 85.162%, specificity of 84.883%, F-measure of 85.111%, and AUC of 85.283%. The ANN-CL method continues this trend of improvement, attaining an accuracy of 90.255%, precision of 88.715%, recall of 88.816%, specificity of 88.537%, F-measure of 88.765%, and AUC of 88.937%. SmartCheck excels further, reaching an accuracy of 93.909%, precision of 92.369%, recall of 92.470%, specificity of 92.191%, F-measure of 92.419%, and AUC of 92.591%. The proposed GR-ANN method outperforms all existing methods, demonstrating a remarkable accuracy of 97.563%, precision of 96.023%, recall of 96.124%, specificity of 95.845%, F-measure of 96.073%, and AUC of 96.245%. This signifies a consistent increase in performance metrics, reflecting the superior efficacy of the GR-ANN method in predicting vulnerabilities associated with the "ARTHM-vulnerability."

Table 2: Comparative analysis of proposed and existing smart contract vulnerability prediction methods for “RENT -vulnerability”

Methods	Metrics (%)					
	Accuracy	Precision	Recall	Specificity	F-measure	AUC
Mythril [12]	78.264	77.586	77.056	77.716	77.320	77.853
PSO-NDS [13]	81.918	81.240	80.710	81.370	80.974	81.507
ASG-TL [14]	85.572	84.894	84.364	85.024	84.628	85.161
ANN-CL [16]	89.226	88.548	88.018	88.678	88.282	88.815
SmartCheck [17]	92.880	92.202	91.672	92.332	91.936	92.469
GR-ANN	96.534	95.856	95.326	95.986	95.590	96.123

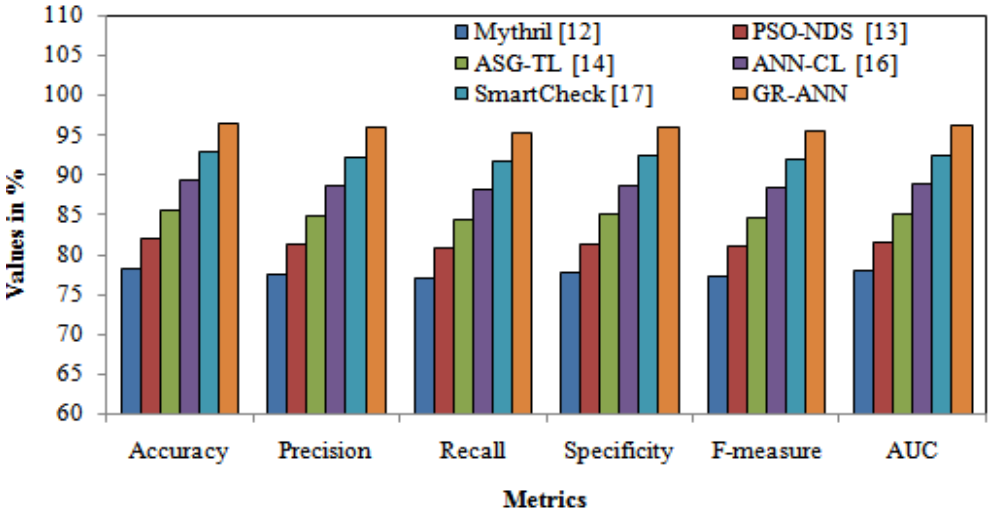

Figure 3: Results comparison for RENT -vulnerability prediction

Table 3 provides a comparative analysis of smart contract vulnerability prediction methods, focusing on the "LE-vulnerability." Commencing with Mythril, the method yields an accuracy of 77.596%, precision of 76.853%, recall of 76.766%, specificity of 77.142%, F-measure of 76.809%, and AUC of 77.625%. PSO-NDS exhibits improvements across all metrics, achieving an accuracy of 81.250%, precision of 80.507%, recall of 80.420%, specificity of 80.796%, F-measure of 80.463%, and AUC of 81.279%. Advancing to ASG-TL, the method continues the upward trajectory with an accuracy of 84.904%, precision of 84.161%, recall of 84.074%, specificity of 84.450%, F-measure of 84.117%, and AUC of 84.933%. The ANN-CL method demonstrates further enhancements, attaining an accuracy of 88.558%, precision of 87.815%, recall of 87.728%, specificity of 88.104%, F-measure of 87.771%, and AUC of 88.587%. SmartCheck excels, reaching an accuracy of 92.212%, precision of 91.469%, recall of 91.382%, specificity of 91.758%, F-measure of 91.425%, and AUC of 92.241%. The proposed GR-ANN method outperforms existing methods, showcasing a significant increase in performance metrics. Specifically, it achieves an accuracy of 95.866%, precision of 95.123%, recall of 95.036%, specificity of 95.412%, F-measure of 95.079%, and AUC of 95.895%. This emphasizes the superior efficacy of the GR-ANN method in predicting vulnerabilities associated with the "LE-vulnerability."

Table 3: Comparative analysis of proposed and existing smart contract vulnerability prediction methods for “LE-vulnerability”

Methods	Metrics (%)					
	Accuracy	Precision	Recall	Specificity	F-measure	AUC
Mythril [12]	77.596	76.853	76.766	77.142	76.809	77.625
PSO-NDS [13]	81.250	80.507	80.420	80.796	80.463	81.279
ASG-TL [14]	84.904	84.161	84.074	84.450	84.117	84.933
ANN-CL [16]	88.558	87.815	87.728	88.104	87.771	88.587
SmartCheck [17]	92.212	91.469	91.382	91.758	91.425	92.241

GR-ANN	95.866	95.123	95.036	95.412	95.079	95.895

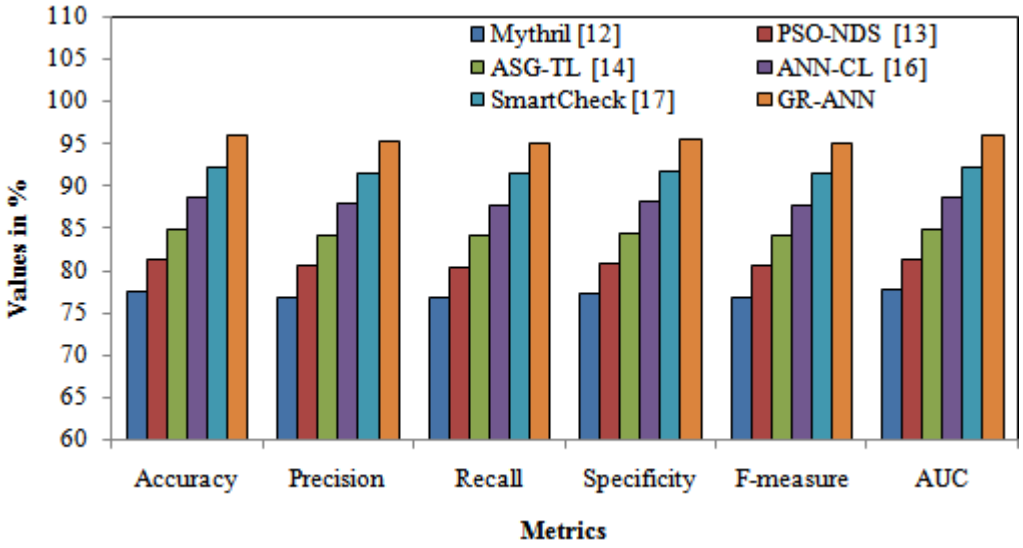

Figure 4: Results comparison for LE-vulnerability prediction

Table 4 presents a comprehensive comparative analysis of smart contract vulnerability prediction methods, specifically focusing on the "TimeO-vulnerability." The evaluation considers key performance metrics, including accuracy, precision, recall, specificity, F-measure, and AUC, providing insights into the effectiveness of each method. Initiating with Mythril, the method exhibits an accuracy of 77.755%, precision of 77.594%, recall of 77.192%, specificity of 76.966%, F-measure of 77.392%, and AUC of 77.187%. Subsequently, PSO-NDS demonstrates incremental improvements across all metrics, achieving an accuracy of 81.409%, precision of 81.248%, recall of 80.846%, specificity of 80.620%, F-measure of 81.047%, and AUC of 80.841%. Advancing to ASG-TL, the method maintains the positive trend with an accuracy of 85.063%, precision of 84.902%, recall of 84.500%, specificity of 84.274%, F-measure of 84.701%, and AUC of 84.495%. The ANN-CL method further enhances performance, attaining an accuracy of 88.717%, precision of 88.556%, recall of 88.154%, specificity of 87.928%, F-measure of 88.355%, and AUC of 88.149%. SmartCheck excels across all metrics, achieving an accuracy of 92.371%, precision of 92.210%, recall of 91.808%, specificity of 91.582%, F-measure of 92.009%, and AUC of 91.803%. Notably, the proposed GR-ANN method outperforms existing methods, demonstrating significant improvements in performance metrics. Specifically, it achieves an accuracy of 96.025%, precision of 95.864%, recall of 95.462%, specificity of 95.236%, F-measure of 95.663%, and AUC of 95.457%. This underscores the superior efficacy of the GR-ANN method in predicting vulnerabilities associated with the "TimeO-vulnerability."

Table 4: Comparative analysis of proposed and existing smart contract vulnerability prediction methods for “TimeO-vulnerability”

Methods	Metrics (%)					
	Accuracy	Precision	Recall	Specificity	F-measure	AUC
Mythril [12]	77.755	77.594	77.192	76.966	77.392	77.187
PSO-NDS [13]	81.409	81.248	80.846	80.620	81.047	80.841
ASG-TL [14]	85.063	84.902	84.500	84.274	84.701	84.495
ANN-CL [16]	88.717	88.556	88.154	87.928	88.355	88.149
SmartCheck [17]	92.371	92.210	91.808	91.582	92.009	91.803
GR-ANN	96.025	95.864	95.462	95.236	95.663	95.457

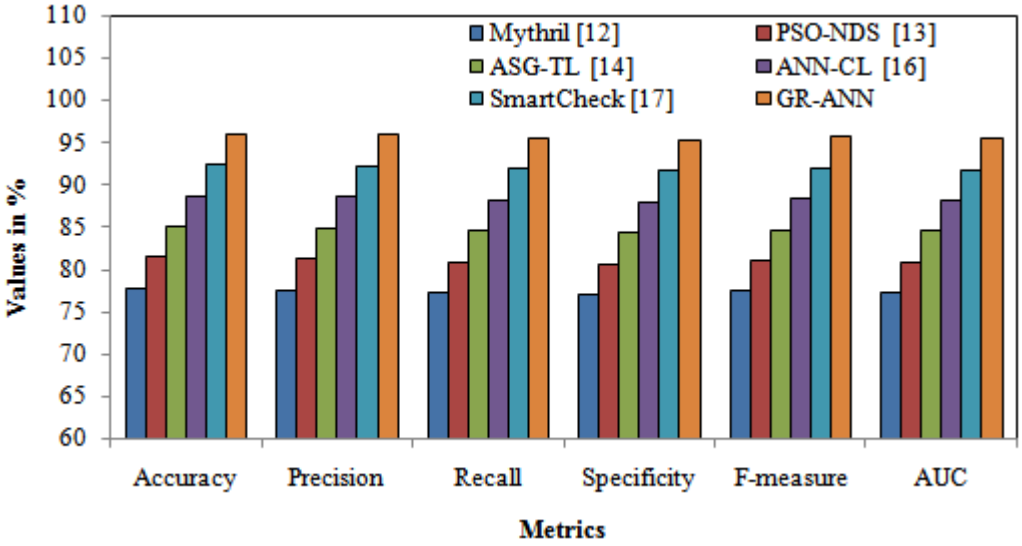

Figure 5: Results comparison for TimeO-vulnerability prediction

5. Conclusion

This study has pioneered a hybrid artificial intelligence and optimization technique for predicting smart contract vulnerabilities in the FinTech domain. Leveraging the modified barnacles mating optimization (MBMO) algorithm, our approach excels in extracting intricate syntactic and semantic features, significantly elevating the precision of vulnerability predictions. Furthermore, we deploy the general regressive artificial neural network (GR-ANN) to forecast vulnerabilities, offering a detailed description of vulnerability types within smart contracts deployed in a cloud environment. The thorough evaluation of this innovative framework is conducted through extensive testing on the ScrawID-real Ethereum smart contract benchmark dataset. The results underscore the remarkable capability and accuracy of our approach in predicting vulnerabilities associated with smart contracts. Analyzing the outcomes, our GR-ANN method demonstrates an impressive average accuracy of 96.467%, showcasing its robust predictive performance. The average precision, standing at 95.714%, attests to the method's ability to accurately identify vulnerabilities without compromising on false positives. Moreover, the average recall of 95.487% highlights the method's efficacy in capturing a substantial proportion of actual vulnerabilities. With an average F-measure of 95.93%, our GR-ANN method strikes a harmonious balance between precision and recall, further solidifying its effectiveness in predicting smart contract vulnerabilities. Overall, the study successfully demonstrates a hybrid AI and optimization technique for predicting vulnerabilities in cloud-based smart contracts within the FinTech industry. The proposed method, validated against the ScrawID-real Ethereum dataset, shows high accuracy and potential for practical application, marking a significant advancement in securing financial technology platforms.

References

[1] Dorsala, M.R., Sastry, V.N. and Chapram, S., 2020. Fair payments for verifiable cloud services using smart contracts. Computers & Security, 90, p.101712.

[2] Tapas, N., Longo, F., Merlino, G. and Puliafito, A., 2020. Experimenting with smart contracts for access control and delegation in IoT. Future Generation Computer Systems, 111, pp.324-338.

[3] Zheng, Z., Xie, S., Dai, H.N., Chen, W., Chen, X., Weng, J. and Imran, M., 2020. An overview on smart contracts: Challenges, advances and platforms. Future Generation Computer Systems, 105, pp.475-491.

[4] Almasoud, A.S., Hussain, F.K. and Hussain, O.K., 2020. Smart contracts for blockchain-based reputation systems: A systematic literature review. Journal of Network and Computer Applications, 170, p.102814.

[5] Xuan, S., Zheng, L., Chung, I., Wang, W., Man, D., Du, X., Yang, W. and Guizani, M., 2020. An incentive mechanism for data sharing based on blockchain with smart contracts. Computers & Electrical Engineering, 83, p.106587.

[6] Patil, A.S., Hamza, R., Hassan, A., Jiang, N., Yan, H. and Li, J., 2020. Efficient privacy-preserving authentication protocol using PUFs with blockchain smart contracts. Computers & Security, 97, p.101958.

[7] Petroni, B.C.A., Gonçalves, R.F., de Arruda Ignácio, P.S., Reis, J.Z. and Martins, G.J.D.U., 2020. Smart contracts applied to a functional architecture for storage and maintenance of digital chain of custody using blockchain. Forensic Science International: Digital Investigation, 34, p.300985.

[8] De Giovanni, P., 2020. Blockchain and smart contracts in supply chain management: A game theoretic model. International Journal of Production Economics, 228, p.107855.

[9] Unal, D., Hammoudeh, M. and Kiraz, M.S., 2020. Policy specification and verification for blockchain and smart contracts in 5G networks. ICT Express, 6(1), pp.43-47.

[10] Zhang, L., Zhang, H., Yu, J. and Xian, H., 2020. Blockchain-based two-party fair contract signing scheme. Information Sciences, 535, pp.142-155.

[11] Xu, Y., Hu, G., You, L. and Cao, C., 2021. A novel machine learning-based analysis model for smart contract vulnerability. Security and Communication Networks, 2021, pp.1-12.

[12] Liu, Z., Qian, P., Wang, X., Zhu, L., He, Q. and Ji, S., 2021. Smart contract vulnerability detection: from pure neural network to interpretable graph feature and expert pattern fusion. arXiv preprint arXiv:2106.09282.
[13] Zhang, L., Wang, J., Wang, W., Jin, Z., Zhao, C., Cai, Z. and Chen, H., 2022. A novel smart contract vulnerability detection method based on information graph and ensemble learning. Sensors, 22(9), p.3581.
[14] Qian, S., Ning, H., He, Y. and Chen, M., 2022. Multi-Label Vulnerability Detection of Smart Contracts Based on Bi-LSTM and Attention Mechanism. Electronics, 11(19), p.3260.
[15] Griggs, K.N., Ossipova, O., Kohlios, C.P., Baccarini, A.N., Howson, E.A. and Hayajneh, T., 2018. Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. Journal of medical systems, 42, pp.1-7.
[16] Huang, J., Zhou, K., Xiong, A. and Li, D., 2022. Smart contract vulnerability detection model based on multi-task learning. Sensors, 22(5), p.1829.
[17] Zhang, L., Chen, W., Wang, W., Jin, Z., Zhao, C., Cai, Z. and Chen, H., 2022. Cbgru: A detection method of smart contract vulnerability based on a hybrid model. Sensors, 22(9), p.3577.
[18] Xing, C., Chen, Z., Chen, L., Guo, X., Zheng, Z. and Li, J., 2020. A new scheme of vulnerability analysis in smart contract with machine learning. Wireless Networks, pp.1-10.
[19] Oliva, G.A., Hassan, A.E. and Jiang, Z.M., 2020. An exploratory study of smart contracts in the Ethereum blockchain platform. Empirical Software Engineering, 25, pp.1864-1904.
[20] Wang, Z., Dai, W., Choo, K.K.R., Jin, H. and Zou, D., 2020. FSFC: An input filter-based secure framework for smart contract. Journal of Network and Computer Applications, 154, p.102530.