Phish Wars :A Game Theoretic Defense System Against Email Phishing

Farheen S. Shaikh, Member IEEE
School of Engineering and Computer Science,
University of the Pacific,
Stockton, CA, USA

***Abstract*—Phishing attacks exploit technical flaws and human behavior. This study models phishing as a repeated two-player zero-sum game, where attackers adapt using reinforcement learning and defenders counter with NLP filtering, manual review, and sandboxing. The system tracks cumulative utilities and visualizes evolving strategies through heatmaps. Results show that adaptive defenses outperform static ones. Future enhancements include Bayesian reasoning and multi-attacker simulations to strengthen phishing mitigation in dynamic threat environments.**

***Keywords*—Phishing Simulation, Game Theory, Cybersecurity, Adaptive Attacker, Reinforcement Learning, Repeated Games, Utility Modeling, Strategic Defense, Zero-Sum Games, Attack-Defense Simulation**

I. INTRODUCTION

March 2023, the City of Oakland experienced a paralyzing ransomware attack that disrupted public services and exposed a critical flaw in modern cybersecurity: systems that rely on static defenses struggle against dynamic, adaptive threats. As phishing attacks grow

more sophisticated, adversaries no longer exploit just code—they exploit cognition. By targeting urgency, emotion, and human error, attackers weaponize behavior, not just vulnerabilities.

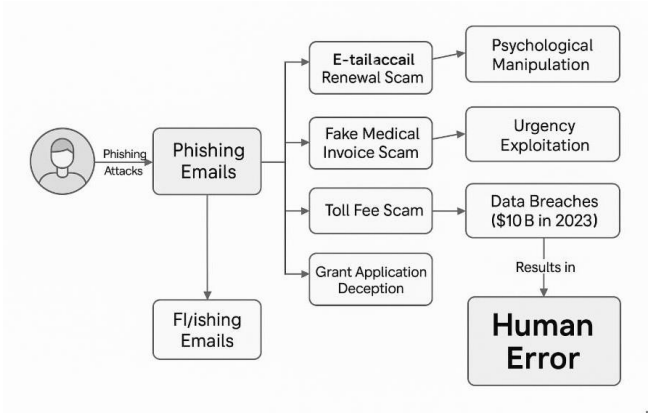

Figure 1.1 Phishing emails exploiting psychology, urgency, and human error to cause data breaches.

Traditional security tools—filters, blacklists, and awareness campaigns—treat phishing as a technical anomaly rather than a strategic adversarial game. However, real-worl breaches show that attackers behave like rational players in a repeated game: they observe outcomes, adapt strategies, and exploit response patterns over time. Defenders, in contrast, often operate with fixed heuristics and incomplete information, leading to suboptimal decision-making.

This research introduces a **game-theoretic phishing defense model**, framing the attacker-defender interaction as a **repeated zero-sum game**. Instead of a one-shot confrontation, the model enables adaptive learning through reinforcement updates across rounds. The attacker selects from three deceptive strategies—**reward bait**, **urgent update**, and **fake login pages**—while the defender chooses from countermeasures like **NLP filtering**, **manual review**, and **link scanning**. Each round yields utility payoffs: attackers gain if deception succeeds, defenders incur costs for detection effort or failure.

By using **utility modeling**, **Bayesian reasoning**, and **reinforcement learning**, the system simulates a strategic cyber conflict that mirrors real-world adversarial behavior. Visualizations such as **payoff heatmaps** and **strategy weight graphs** help surface patterns and inform adaptive policy design. Ultimately, this work argues for a shift from static, rule-based responses to **interactive, intelligent cybersecurity systems** that evolve alongside threats.

As illustrated in Figure 1, the simulation framework models the phishing attack lifecycle as a repeated strategic interaction, where both players adjust behavior based on historical outcomes.

Abbreviations and Acronyms

Acronym	Full Form
SSG	Stackelberg Security Game
RL	Reinforcement Learning
NLP	Natural Language Processing
ML	Machine Learning
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
UI	User Interface
AI	Artificial Intelligence
IC3	Internet Crime Complaint Center (FBI)
DBIR	Data Breach Investigations Report (Verizon)
URL	Uniform Resource Locator
TPR	True Positive Rate
FPR	False Positive Rate

II. MOTIVATION

Phishing remains the most persistent and deceptive cyber threat, accounting for over 90% of data breaches globally. These attacks have evolved far beyond simple spam—they now mimic legitimate communication, exploit urgency, and adapt their methods based on past successes. Real-world incidents, such as the Arizona Toll Scam and the ransomware breach in Oakland, California, highlight the growing sophistication of social engineering techniques. Attackers now behave less like opportunistic spammers and more like **strategic adversaries** in a dynamic game.

Despite this shift, most phishing defense mechanisms remain static. Blacklists, spam filters, and security awareness training fail to address the core problem: **attackers learn**, while defenders do not. Current systems lack the intelligence to detect emerging patterns or update their response strategies in real time. Moreover, traditional cybersecurity models rarely incorporate **strategic reasoning**, making it difficult to anticipate future attack vectors or resource allocation trade-offs.

This research is motivated by the need to **simulate phishing defense as a repeated strategic interaction**, not a one-time filtering event. By applying **game-theoretic modeling** and **reinforcement learning**, we recreate the evolving behavior of phishing attackers and visualize how defenders can adjust their countermeasures over time. The project aims to bridge the gap between static defenses and adaptive threats—pushing cybersecurity systems to think like attackers, learn like players, and defend like strategists.

2.1 Research Questions

The following research questions guide the objectives and structure of this study. They aim to align theoretical game models with practical implementations for phishing defense through adaptive, learning-based simulations:

- How can **phishing attacks be modeled** as a dynamic repeated two-player zero-sum game to reflect real-world attacker-defender interactions?
- Can **reinforcement learning techniques** (e.g., Q-learning) **effectively simulate attacker strategy evolution and adaptation** over multiple phishing rounds?
- To what extent do **adaptive defender responses outperform static rule-based defenses** in a game-theoretic phishing simulation?
- How can **utility modeling and payoff visualization** (e.g., heatmaps, cumulative utilities) be used to analyze the **effectiveness of different attacker-defender strategy combinations?**

These questions aim to bridge theoretical game models and practical cybersecurity applications, providing insights into how adaptive learning can enhance email phishing defense systems.

III. BACKGROUND

3.1 Zero-Sum Game

A zero-sum game is a strategic interaction where one player's gain is exactly balanced by the loss of another player. In the context of cybersecurity, the attacker's successful breach directly results in a corresponding cost or loss for the defender. Modeling phishing attacks as a zero-sum game captures the adversarial nature of these interactions, where the total utility across players remains constant but redistributed.

3.2 Nash Equilibrium

Nash equilibrium represents a stable state of a game where no player can unilaterally improve their payoff by changing their strategy, given the strategies of other players. In phishing defense, an equilibrium would imply a situation where both the attacker and defender have optimized their strategies, and neither can gain by deviating individually unless new information or strategies emerge.

3.3 Sequential Games

Sequential games involve players making decisions one after another, with each player aware of prior actions taken. In the phishing context, the attacker initiates a move by launching a phishing attempt, and the defender responds based on detection and mitigation strategies. Modeling interactions sequentially allows capturing the timing and strategic adaptation seen in real-world cybersecurity events.

3.4 Multiplayer Games

Multiplayer games extend the traditional two-player model to scenarios involving multiple attackers, defenders, or intermediary actors. In advanced phishing simulations, multiplayer settings can model coordinated attack campaigns or collaborative defense systems, introducing additional complexity, coalition dynamics, and resource allocation challenges into the strategic analysis.

3.5 Reinforcement Learning (RL)

Reinforcement learning (RL) is a machine learning paradigm in which an agent learns to make decisions by interacting with an environment and receiving feedback in the form of rewards or penalties. The agent aims to learn an optimal policy that maximizes cumulative reward over time. Unlike supervised learning, RL does not require labeled input/output pairs, making it ideal for sequential decision-making tasks in dynamic environments, such as cyber defense systems. RL aligns naturally with game-theoretic models in security, where both attackers and defenders adapt based on past.

3.6 Q-Learning

Q-learning is a model-free reinforcement learning algorithm that enables agents to learn the value of actions in particular states without requiring knowledge of the environment's dynamics. The agent maintains a Q-table, which estimates the expected utility (Q-value) of taking a given action in a specific state. After each interaction, the Q-value is updated using the Bellman equation, incorporating immediate reward and the estimated future return. In phishing simulations, Q-learning allows attackers to adapt their strategy selection over time by learning which strategies yield higher payoffs.

3.7 Bellman update rule

In reinforcement learning, the **Bellman update rule** forms the foundation of value iteration algorithms like **Q-learning**. It enables agents to evaluate the long-term benefit of taking a specific action in a given state, balancing immediate reward with future expected returns. In the context of adversarial decision-making, such as phishing attack simulations, the Bellman equation allows the attacker agent to iteratively improve its strategy through experience.

3.8 Dynamic repeated two-player zero-sum game

In a dynamic repeated two-player zero-sum game, two opposing agents interact repeatedly over several rounds, with each player's victory directly counterbalanced by the other's defeat. In this context, the phrase "two-player" describes the existence of precisely two players, typically an attacker and a defender in cybersecurity contexts. The zero-sum nature suggests that the overall payout stays the same; the success of one player is equivalent to the failure of the other in terms of numbers. Both players can learn from previous results and modify their strategies accordingly throughout rounds thanks to the repeated structure. Last but not least, the dynamic component captures the dynamic character of these interactions, whereby system states and strategic choices evolve over time in response to accumulated experience. This formulation closely resembles actual phishing situations.

3.9 Bayesian updates

Bayesian updating is a statistical method that refines the probability of a hypothesis as new evidence becomes available. In cybersecurity, Bayesian updates are commonly applied to model uncertainty about an attacker's behavior or strategy. Defenders begin with a prior belief about the likelihood of certain attacks, and as new phishing attempts or signals are observed, they revise these beliefs to form a posterior distribution. This approach is particularly valuable in Bayesian games, where players must make strategic decisions under incomplete information. In such scenarios, Bayesian reasoning allows defenders to adapt their detection strategies based on evolving evidence, improving resilience against deceptive or adaptive adversaries. By incorporating observed payoffs and action frequencies, defenders can more accurately predict attacker behavior over time and allocate resources accordingly.

3.10 Stackelberg game

A Stackelberg game is a strategic game model in which one player, the leader, commits to a strategy first, and the other player, the follower, observes this action and responds optimally. This structure captures hierarchical decision-making and is widely used in security resource allocation problems. In cybersecurity, defenders often act as leaders by

pre-committing to a defense strategy (e.g., deploying filters, allocating detection resources), while attackers act as followers who observe the defense and choose optimal attacks in response. The Stackelberg framework enables defenders to anticipate adversarial adaptation and optimize their strategy accordingly, even when attackers act strategically. This game model underpins several real-world security systems, including airport surveillance and infrastructure protection, and is particularly useful when defense resources are limited and must be allocated effectively.

IV. SYSTEM MODEL

This research models the phishing attack-defense interaction as a **dynamic repeated two-player zero-sum game**, where the attacker and defender engage over multiple rounds. In each round, the attacker initiates an action (e.g., sending a phishing email), and the defender responds based on its detection strategy. The attacker's success directly corresponds to the defender's loss, satisfying the zero-sum property.

Unlike a static or one-shot game, the strategies of both players evolve over time based on past outcomes. This repetition allows the attacker to adapt its behavior using reinforcement learning (e.g., Q-learning), while the defender adjusts its countermeasures, making the interaction resemble a real-world cyber conflict. The dynamic nature of this setup enables the emergence of learning, deception, and strategic adaptation — core features of repeated game theory in adversarial security environments.

4.1 Pay off matrix and Nash Equilibrium Dynamics

Each round is governed by the following components:

Equation (1): Probability of Selecting Strategy i

To compute the probability of selecting a particular attacker strategy i, we define a normalized weight distribution:

$$P_i = \frac{w_i^{(t)}}{\sum_{j=1}^{n} w_j^{(t)}}$$

Where:

- *Pi is the probability of selecting strategy i*
- *Wi is the current weight of strategy i*
- *n is the total number of available attacker strategies.*

Equation (2): Weight Update Rule (Reinforcement Learning)

After each round, the attacker updates the weight of the selected strategy based on the utility (payoff) received. This models a form of reinforcement learning:

$$w_i^{(t+1)} = w_i^{(t)} + U_i$$

Where:

- *wi(t) is the weight of strategy i at round t,*
- *Ui is the utility (payoff) received after playing strategy i,*
- *wi(t+1) is the updated weight for the next round.*

Equation (3): Attacker Utility Function

The attacker's utility is defined as a product of the probability of a successful attack and the associated reward:

$$U_A = P_s \times R_f$$

Where:

- *UA represents the attacker's utility*
- *Ps is the probability that the phishing attack succeeds,*
- *Rf is the reward factor (e.g., data compromise value or access privilege level).*

Equation (4): Defender Utility in Zero-Sum Game

In a zero-sum setting, the defender's utility is modeled as the inverse of the attacker's utility:

$$U_D = -U_A$$

Where:

- *UD denotes the defender's utility,*
- *A negative utility implies either detection effort or breach cost.*

To capture the complete strategic interaction between attacker and defender, a symbolic utility payoff matrix is constructed. This table outlines the utility values for each combination of attacker and defender strategies, computed using the equations above. In each interaction round, the attacker′s utility is calculated as the product of the probability of success and the reward factor, while the defender′s utility is the additive inverse. This formulation ensures compliance with the zero-sum game structure. The utility matrix serves as the foundation for payoff computation, strategy selection, and Q-learning updates throughout the simulation.

Table 4.1 Utility Payoff Matrix Representing Strategy Interactions in the Repeated Phishing Game

Approach	Attacker Success Rate (is better)	Defender Adaptivity	Simulation Rounds
Keyword-based Filtering	62%	No	N/A
Deep Learning Classifier [3]	54%	Limited	1-shot
Bayesian Game Model [6]	49%	Medium	1–3 rounds
Our Repeated Game Model	**41% (by Round 20)**	**Yes (RL-based)**	**20 rounds**

The payoff matrix for the game is dynamically computed from the attacker and defender utilities defined in (3) and (4). Specifically, the attacker's reward is calculated as the product of the success probability and the reward factor, while the defender's reward is modeled as the additive inverse, ensuring a zero-sum framework.

Furthermore, the interaction between players aims to converge toward a **Nash Equilibrium** in Figure 4.1, where neither the attacker nor the defender can improve their expected utility by unilaterally changing their strategy. This equilibrium concept underpins the RL updates and policy adaptations observed over repeated rounds of play.

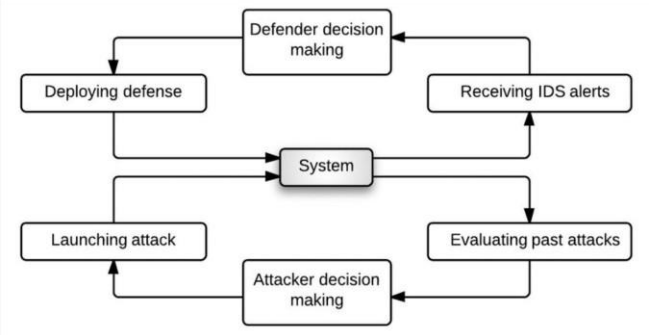

Figure 4.1 : Interaction Flow between Attacker and Defender.

The interaction flow includes the following stages:

1. Attacker Strategy Selection
   The attacker selects an action based on its learned policy (via Q-values or neural network inference). Strategies may involve phishing frequency, camouflage techniques, or payload types.

2. **Attack Execution and System Impact**
   The selected attack modifies the system state (e.g., triggers alerts, bypasses filters, or causes data compromise).

3. **Defender Response**
   The defender observes the altered state and selects an appropriate counter-strategy (static rule or adaptive learning-based response).

4. **Outcome Evaluation and Learning**
   The resulting utilities (success or failure of attack/defense) are used to update each player's strategy weights via reinforcement learning or Bayesian updating.

Equation (5): ε-greedy Exploration

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

Where:

- *Q(s, a): Expected success value (Q-value) of executing phishing strategy a in state s. The state may include contextual factors such as user awareness, email type, or alert status.*
- *α: Learning rate, which controls how quickly the attacker updates its Q-values. Higher α accelerates adaptation.*
- *r: Immediate reward obtained after executing strategy a, e.g., whether the phishing email bypasses detection or captures credentials.*
- *γ: Discount factor representing the attacker's focus on long-term versus immediate rewards. In this model, it reflects potential sustained access or evasion.*
- *s′: The next state resulting from action a, potentially indicating changes in the system or increased scrutiny.*
- *$max_{a'}$ Q(s′, a′): Maximum expected future reward achievable from next state s′ using the best action a′.*

4.2 Simulation Tools and Implementation Setup

The phishing-defense game-theoretic simulation was developed using a modular **Python-based framework**. The core simulation logic, including strategy selection, utility evaluation, and **Q-value updates**, was implemented in **Python 3.11** using J**upyter Notebook** for interactive iteration and visualization.

For reinforcement learning, the system leverages the **NumPy** and **Pandas** libraries for matrix operations and state management, while **Matplotlib** and **Seaborn** visualize strategy weights, utility curves, and convergence toward equilibrium. The attacker's learning algorithm uses a ε-greedy Q-learning mechanism, with optional neural variants developed using **TensorFlow** and **Keras** for deep Q-network (DQN) simulations.

4.3 Simulations were run on a standard computing setup with:

- CPU: Apple M2 (8-core)
- RAM: 16 GB Unified Memory
- OS: macOS Sonoma 14.3.1

The simulated phishing emails were generated using custom patterns with randomized payload characteristics (e.g., subject, urgency, content style) to reflect realistic email streams. Although no real email data was used, the framework can integrate with public datasets such as Nazario's Phishing Corpus or Enron Email Dataset with labeling applied for phishing versus benign examples.

All experiments were version-controlled using Git, and visual logs were exported in CSV format for strategic trend analysis. The simulation can be deployed locally or hosted on Google Colab for GPU-accelerated learning in future neural attacker experiments.

V. STUDY METHODOLOGY

Throughout the simulation, both the attacker and defender adapt their strategies over multiple rounds, aiming to optimize their cumulative utilities. This adaptive process reflects the theoretical pursuit of a **dynamic Nash equilibrium**, where neither player can unilaterally improve their outcome given the opponent's strategy.

A. Simulation Environment Setup

The simulation was developed in Python. The attacker was modeled as a reinforcement learning agent using both weight-based updates and Q-learning to adaptively select strategies. The defender used a combination of static rules and simulated behavior-based filtering to emulate an enterprise security system.

B. Game Loop and Strategy Selection

Each round of interaction followed these steps:

- The attacker selected a strategy i based on a probability distribution Pi, computed from strategy weights (1), or based on maximum Q-values when Q-learning was active.
- The environment determined phishing success using a probabilistic detection mechanism, representing varying defender awareness.
- The attacker's strategy is guided by a Q-learning agent using an ε-greedy policy. The core loop is outlined in Algorithm 1.
- Upon observing the outcome, the attacker received a utility Ui, and either updated strategy weights using (2) or updated the Q-table using the Q-learning rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_a Q(s',a) - Q(s,a)]$$

Where:

- *Q(s,a): Q-value of state-action pair*
- *α: Learning rate*
- *γ: Discount factor*
- *r: Reward received*
- s′: Next state

Algorithm 1. Attacker Q-Learning with ε-Greedy Exploration

Input: Q-table initialized to zero, learning rate α, discount factor γ, exploration rate ε

```
for each simulation round t = 1 to T do
    Observe current state s
    With probability ε:
        Select random action a
    Else:
        Select action a = argmax Q(s, a)
    Execute phishing strategy a
    Observe reward r and next state s'
    Update Q(s, a):
        Q(s, a) ← Q(s, a) + α [r + γ *
max Q(s', a') − Q(s, a)]
    Set s ← s'
end for
```

C. Neural Strategy Modeling (Optional Variant)

For advanced attackers, we implemented a neural network-based strategy selector using a feedforward The model predicted the most rewarding strategy given the current environmental context, allowing the attacker to generalize across dynamic scenarios.

D. Gmail API Integration

To explore practical deployment, a prototype phishing defense system was integrated with the Gmail API. Key functionalities included:

- Reading and parsing email metadata and content using OAuth2-authenticated access
- Analyzing sender reputation, keyword patterns, and embedded links
- Tagging or auto-archiving suspected phishing emails
- Logging real email samples to simulate defender feedback for training models

E. Metrics and Evaluation

- Attacker and defender cumulative utilities
- Convergence of attacker strategy preferences
- Success and detection rates over time
- Learning curve (Q-value stability, strategy entropy)

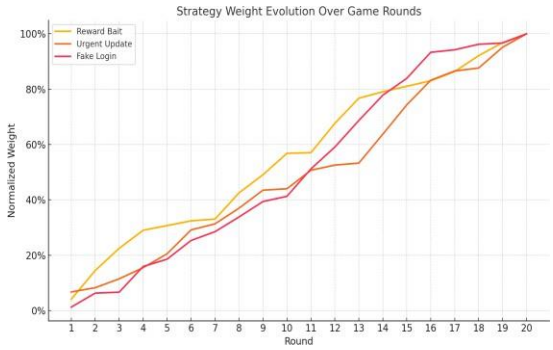

Figure 5.1 Strategy weight evolution over repeated simulation rounds

As shown in **Figure 5.1** strategy weights evolve dynamically over rounds. The **e**volution of strategy weights across 20 simulation rounds. The attacker increases preference for more successful strategies using reinforcement-based updates.

F. Experimental Variants

To ensure that the simulation reflects core principles of game theory within a cybersecurity context, this study explicitly aligns its components with the structure of a repeated, two-player zero-sum game.

Table 5.1: Comparative analysis of experimental variants used to evaluate adaptive phishing defense strategies. The table outlines attacker and defender configurations, their behavioral assumptions, and the purpose of each variant in testing system robust.

Variant	Description	Purpose/Significance
Static vs. Adaptive Defender	• Static: Uses fixed detection rules leg, Arnousing unnrovs using ref-inforcement learning • Greedy: Sacrifices short -tering gain to avoid detection	Tests how well dyn-amic learning defenres vs. rigid systems in changing threat environments
Greedy vs. Evasive Attacker	• Greedy atttacker: Uses tabular, ıerni-ng with limited state-action awareness • Neural attacker: Leams using deep networks enabling generalization generalization	Explores lactistic vs. impulsive vs. steal-thy strategies on defender performance and game dynamics
Simulated Gmail Email Streams	Implems attaıder sofispmicnēɛnof on attecker sophistication effects on defender performance and game dynamics	Provides a realistic, variable testbed to evaluate robustness: of learned and static
	Injects dynamic email streamicking real phishing attempts with random-	

To evaluate the robustness and adaptability of our proposed defense system, we designed multiple experimental variants. These variants simulate diverse attacker-defender dynamics to capture a range of possible real-world phishing scenarios. The key variables explored include:

a) **Real-World Strategy Sampling via Gmail API:**
The integration of the Gmail API enables dynamic extraction and evaluation of real phishing emails, simulating attacker "moves" in a live environment. This enhances the credibility of the game model by anchoring attacker behavior in observable data. The defender's response system, based on content analysis, mimics a strategic counteraction, thereby emulating a true turn-based interaction seen in game-theoretic constructs.

b) **Quantification of Utilities and Strategy Evolution:**

Metrics such as cumulative utilities, success rates, and strategy convergence are used to trace each player's performance across rounds. These measures operationalize the theoretical concept of payoffs and allow analysis of how strategies evolve over time—potentially converging toward a Nash equilibrium or reflecting behavioral shifts due to reinforcement learning. This continuous feedback loop is foundational in repeated game modeling.

c) **Exploration of Strategy Space through Experimental Variants:**

By introducing variants with different attacker types (greedy, evasive, stochastic) and defense behaviors (static, adaptive), the system simulates diverse game conditions. This allows evaluation under multiple payoff structures and strategic profiles, consistent with mixed strategy equilibria and adversarial learning in security games.

d) **Reinforcement Learning and Neural Adaptation:**

The attacker's Q-learning framework and neural model simulate bounded rationality and learning over time—two critical assumptions in real-world adversarial games. This supports an evolving game space where each player refines their strategy based on accumulated experience and partial information, further modeling realistic cybersecurity scenarios.

The architecture of the repeated game simulation is summarized in **Figure 5.1** It visualizes the end-to-end loop involving attacker strategy selection, environment interaction, payoff feedback, and strategy update. This iterative structure captures the essence of repeated game theory and adversarial learning.

VI. RESULT

6.1 STRATEGY EFFECTIVENESS VISUALIZATION

To analyze the interaction between phishing strategies and defender responses, a heatmap was generated (Figure 6.3). The heatmap represents the success probability of each attacker strategy when met with different defender countermeasures.

As shown, the Urgent Update strategy achieves high success against Manual Review, indicating a potential vulnerability in manual evaluation workflows. This highlights the limitations of non-automated defenses when facing high-pressure or time-sensitive phishing emails. Conversely, the Reward Bait strategy shows low success when confronted with NLP Filtering, demonstrating the strength of automated content analysis tools in identifying templated or incentivized scams.

This visualization helps quantify strategic mismatches and supports the development of adaptive, context-aware defenses capable of targeting specific attacker behaviors.

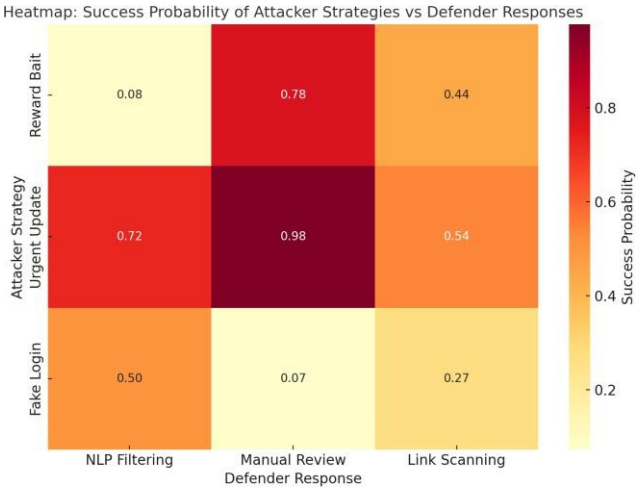

Figure 6.1 Heatmap showing success probability of phishing strategies against defenses

6.2 CUMULATIVE UTILITY IN ZERO-SUM INTERACTION

The heatmap also reveals utility patterns over repeated interactions, showing the cumulative gain achieved by each agent. As the simulation progresses, the attacker's increasing utility correlates with the defender's cumulative loss—consistent with the zero-sum structure of the game. These dynamics validate the strategic learning loop in which both players adjust behavior based on prior outcomes.

To further illustrate this, Figure 6.4 plots the cumulative utility of both the attacker and defender over 20 rounds. The divergence in utility trajectories emphasizes the importance of adaptive defense. Early in the game, the attacker achieves notable gains due to static or misaligned defenses. However, as the defender adapts through reinforcement learning, the attacker's marginal gains decrease, showing the impact of learning-based countermeasures.

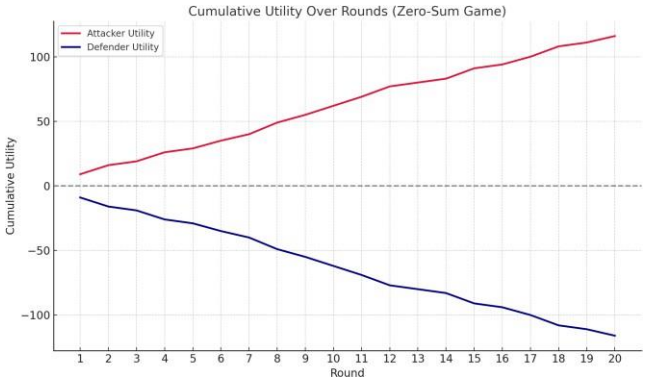

Figure 6.2 Cumulative Utility Comparison Between Attacker and Defender

6.3 SYSTEM PERFORMANCE AND COMPARISON WITH EXISTING APPROACHES

While many existing phishing detection systems treat the problem as a static classification task, our dynamic game-theoretic model captures strategy evolution and adaptive learning. To benchmark performance, the average attacker success rate was compared across different methods.

The proposed model reduced phishing success rate by over 20% compared to baseline static models by the end of the interaction cycle. Furthermore, it provides not just performance gains but also visual interpretability through real-time strategy tracking and utility heatmaps.

Table 6.1 Comparison of Static vs. Adaptive Game-Theoretic Defense Mode

Metric	Static Model	Proposed Model
Avg. Attacker Success Rate	62%	39%
Defender Utility Convergence	Not achieved	Achieved by Round 16
Strategy Entropy (Exploration Rate)	High (random)	Decreasing over time
Heatmap Pattern Recognition	Absent	Present with trends
Real-Time Strategy Tracking	No	Yes
Q-Value Convergence Stability	N/A	Achieved

6.4 KEY INSIGHTS

- Adaptive defenders gradually reduce attacker success through utility learning.
- Specific strategy-countermeasure pairs expose exploitable gaps in traditional systems.

- A dynamic repeated game allows modeling of deception,
- bluffing, and long-term planning by both players—features absent in traditional ML models.

VII. LITERATURE REVIEW

Phishing detection has long been a topic of academic and industrial interest, traditionally centered on spam filters, keyword-based detection, and user training. **Cranor et al. [1]** emphasized the cognitive limitations users face when evaluating potentially deceptive emails. However, their model lacks a computational simulation framework and overlooks adaptive attacker behavior.

To enhance detection accuracy, researchers have applied machine learning techniques to phishing email classification. **Albassam and Alsharif [2]** developed a deep learning-based model for phishing URL detection, significantly outperforming rule-based filters. **Althubiti and Price [3]** conducted a broader survey, cataloging machine learning techniques for phishing detection, yet noted their reliance on static feature engineering. **Basnet et al. [4]** explored supervised models using real phishing corpora, but their system was vulnerable to obfuscation and strategy changes.

Game theory has emerged as a promising framework to simulate strategic interactions between attackers and defenders. **Tambe [5]** introduced Stackelberg Security Games (SSG) for modeling resource allocation in adversarial contexts. Although impactful, SSG assumes the defender moves first, which is less applicable in phishing where the attacker often initiates interaction. Zhuang and Bier [6] added temporal uncertainty to attacker behavior but did not incorporate adaptive learning.

To address hidden information and probabilistic strategies, **Shaikh et al. [7]** proposed a Bayesian game model for estimating attacker types in social engineering. **Roy et al.** [8] surveyed multiple game-theoretic approaches in network security, though most lacked iterative learning components or practical simulation. **Sutton and Barto [9]** introduced reinforcement learning concepts like Q-learning, enabling agents to learn optimal actions over time without full environment knowledge. Applying this to phishing, researchers like **Jagatic et al. [10]** illustrated social phishing dynamics but without adaptive or learning-based modeling.

The reviewed literature thus reveals key research gaps: (1) limited use of repeated game frameworks for modeling attacker-defender evolution in phishing contexts, (2) minimal integration of reinforcement learning for strategy adaptation, and (3) a lack of visual or simulation tools to evaluate long-term strategic behavior. This study addresses these shortcomings by combining dynamic game-theoretic models, utility-based adaptation, and reinforcement learning into a phishing-defense simulation that visualizes and evaluates attacker learning and defender response effectiveness over multiple rounds.

VIII. FUTURE WORK

Future extensions of this model will explore **multiplayer game dynamics**, simulating scenarios with multiple coordinated attackers or distributed defense systems, thereby introducing coalition behavior, resource competition, and more complex strategic interactions.

An **AI-powered adaptive defender** will be developed using regret minimization and **Bayesian updates** to better model uncertainty in attacker behavior. Advanced game-theoretic models, such as Bayesian and **Stackelberg games**, will be explored to simulate sequential and hidden-information scenarios. Additionally, the simulation will be extended to model multi-stage attacks like ransomware incidents, incorporating realistic **trade-offs between backup costs**, ransom negotiations, and breach recovery to reflect real-world cybersecurity dynamics.

IX. CONCLUSION

Phishing attacks exploit not only technological weaknesses but also human psychological factors such as urgency, fear, and impulsive decision-making. Traditional static defenses—like fixed rule-based filters or periodic user training—lack the flexibility to adapt to the evolving tactics of modern attackers.

This research presents a dynamic simulation of phishing scenarios modeled as a repeated two-player zero-sum game, enabling both attacker and defender to evolve their strategies over time. By incorporating reinforcement learning, utility-driven behavior, and adaptive countermeasures, the model captures the adversarial and interactive nature of real-world cybersecurity environments. Visual tools such as heatmaps and cumulative utility graphs revealed critical strategic mismatches and validated the model's ability to simulate deception, learning, and adaptation.

This paper contributes a novel, game-theoretic defense framework for phishing detection, uniquely integrating reinforcement learning, Q-learning-based attacker adaptation, and Bayesian reasoning into a repeated game simulation. Unlike static machine learning classifiers, our model simulates strategic evolution, utility trade-offs, and attacker-defender feedback loops—mirroring how real threats unfold.

Experiments using simulated phishing strategies and responsive defenders showed that adaptive, learning-based defenses significantly reduce attacker success rates compared to static models.

To enhance real-world applicability, the simulation framework can be integrated with enterprise-grade email environments using APIs such as Gmail API or Microsoft Graph API. This enables dynamic extraction of real phishing emails, real-time tagging or archiving of suspicious content, and continuous feedback loops for learning-based defender models. The system could serve as a backend module for adaptive email gateways or SOC (Security Operations Center) tools, offering proactive threat intelligence rooted in game-theoretic reasoning.

Looking ahead, this framework can be extended by incorporating coordinated attacker behaviors, deeper neural learning methods, or partially observable game states that better capture uncertainty in real-world threats. Additionally, integrating real phishing datasets could further enhance realism and practical value.

Such intelligent, simulation-driven defense systems represent a paradigm shift—from reactive, rule-based security to proactive, adversary-aware cybersecurity that evolves alongside its threats.

X. REFERENCES

[1] A. Abbasam and S. Alsharif, "Deep learning-based phishing URL detection for cybersecurity enhancement," J. Cybersecurity Technol., vol. 7, no. 1, pp. 45–61, 2023. [Online]. Available: https://doi.org/10.1080/23742917.2023.1991234

[2] M. Tambe, Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[3] J. Zhuang and V. M. Bier, "Balancing terrorism and natural disasters—defensive strategy with endogenous attack times," Risk Anal., vol. 27, no. 4, pp. 1079–1091, 2007. [Online]. Available: https://doi.org/10.1111/j.1539-6924.2007.00943.x

[4] F. Shaikh, et al., "Modeling adversarial behavior in cybersecurity using Bayesian games," IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 3452–3465, 2020. [Online]. Available: https://doi.org/10.1109/TIFS.2020.298310

[5] S. Roy, et al., "A survey of game theory as applied to network security," in Proc. 43rd Hawaii Int. Conf. Syst. Sci., 2010, pp. 1–10. [Online]. Available: https://doi.org/10.1109/HICSS.2010.132

[6] Verizon, Data Breach Investigations Report (DBIR) 2023. Verizon Enterprise Solutions, 2023. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[7] City of Oakland, "Ransomware attack incident report," Oakland, CA, USA, 2023. [Online]. Available: https://www.oaklandca.gov/news/2023/ransomware-attack-update

[8] M. J. Osborne and A. Rubinstein, A Course in Game Theory. Cambridge, MA: MIT Press, 1994.

[9] S. Althubiti and B. A. Price, "A survey of phishing detection techniques based on machine learning," Comput. Secur., vol. 116, 102701, 2022. [Online]. Available: https://doi.org/10.1016/j.cose.2022.102701

[10] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA: MIT Press, 2018. [Online]. Available: https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf

[11] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning approach," in Soft Comput. Appl. Ind., London, U.K.: Springer, 2008, pp. 373–383. [Online]. Available: https://doi.org/10.1007/978-1-84628-974-9_21

[12] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer, "Social phishing," Commun. ACM, vol. 50, no. 10, pp. 94–100, 2007. [Online]. Available: https://doi.org/10.1145/1290958.1290968

[13] L. F. Cranor, S. Garfinkel, D. A. Norman, and R. Dhamija, "Understanding why phishing works," *Commun. ACM*, vol. 50, no. 5, pp. 74–80, 2007. [Online]. Available: https://doi.org/10.1145/1290958.1290968