The Evolutionary Dynamics of Machine Learning Techniques on Spam Filter

Zhuo (Cecilia) Chen & Ferida Mohammed

May 2025

1 Abstract

This paper examines the ongoing battle between spam filters and spammers through the lens of evolutionary game theory. We model this conflict as a strategic interaction where both sides continuously adapt their approaches—spammers by modifying their techniques to evade detection, and filters by updating their classification rules. Using payoff matrices, we analyze Nash equilibria and evolutionarily stable strategies to understand long-term outcomes. We then extend this analysis to spatial games on grid networks, demonstrating how localized interactions influence strategy adoption. Our simulations reveal distinct patterns: cooperative strategies (adding ”good words”) spread sustainably from small clusters, while defecting strategies (avoiding good words) create unstable oscillations. These findings highlight the fundamental instability in spam filtering systems, where neither side can maintain permanent dominance, leading to an endless arms race. Our results suggest that effective spam filters must incorporate adaptive learning mechanisms that account for these evolutionary dynamics.

2 Introduction

2.1 What is Machine Learning?

Machine learning (ML) is a part of artificial intelligence (AI) that focuses on teaching computers to learn from data and make decisions without being directly programmed for every task. Instead of using fixed rules written by a programmer, machine learning systems find patterns in data and use those patterns to make predictions or decisions. This way of learning has become a key part of many technologies today, including voice assistants, recommendation systems, and spam filters [1].

One important type of machine learning is called supervised learning. In supervised learning, the computer is given many examples where both the input (like an email message) and the correct output (like "spam" or "not spam") are known. The model then learns from these examples and can make predictions on new, unseen data. Some commonly used models in supervised learning include decision trees, support vector machines (SVMs), Naive Bayes classifiers, and neural networks [2].

2.2 Why Spam Identification Matters?

Spam detection is one of the earliest and most widely used applications of machine learning. Spam emails waste people's time, take up storage space, and can even cause harm by spreading viruses or tricking users into giving away private information. Studies show that spam makes up more than half of all email traffic worldwide, so having effective spam filters is essential for protecting users and systems [3].

What makes spam filtering difficult is that spammers are always trying new tricks to beat the system. They might change the wording of their messages, add misleading words, or use images instead of text to avoid being detected. This ongoing struggle between spam filters and spammers means that spam detection systems need to be flexible and able to learn from new threats. Machine learning is a good fit for this because it can keep learning and adjusting as new kinds of spam appear [3].

2.3 Machine Learning in Spam Classification

Today, most spam filters use machine learning to sort messages automatically. The process usually starts with gathering a large set of emails that are already labeled as either spam or not spam. These emails are then turned into a format that a computer can understand—usually by counting how often certain words appear. A machine learning model is trained on this data to learn what spam looks like and then used to check new messages.

Two popular models for spam filtering are Naive Bayes and SVMs. Naive Bayes is simple and fast, but not always very accurate. SVMs usually give better results but are slower and more complex [4]. Some filters also use user feedback to improve over time. For example, if a user marks a message as spam, the model can learn from that and avoid making the same mistake again. More recently, researchers have started exploring how to make these filters stronger against spammers who try to fool them on purpose [5].

2.4 This Paper

In this paper, we study the ongoing struggle between spam filters and spammers using tools from game theory. While most spam filters are built using machine learning models trained on past data, we look at the problem as a kind of game where both sides—filters and spammers—keep changing their strategies. We use payoff matrices to model how different strategies perform and analyze which ones are likely to succeed over time. Concepts like Nash Equilibrium and Evolutionarily Stable Strategy (ESS) help us understand what happens when both sides try to outsmart each other.

We also go one step further by introducing spatial games. In these models, filters and spammers are placed on a grid, and each one interacts only with its neighbors. This setup helps us see how strategies spread across a network over time and how local interactions can affect the bigger picture. By combining machine learning with ideas from game theory, we hope to better understand how spam filters can stay ahead in this ongoing battle.

3 Machine Learning as Spam Filters

3.1 Overview

Email spam filtering is one of the most prominent applications of machine learning. Machine learning techniques are generally categorized into three main types: Supervised, Unsupervised, and Reinforcement Learning, each of which has been applied to the task of spam detection [6]. Supervised Learning encompasses models such as Naive Bayes, Support Vector Machines (SVM), Decision Trees, and Neural Networks, all of which rely on labeled datasets to learn patterns in spam and non-spam messages. Unsupervised Learning approaches, including clustering techniques like Hierarchical and Partition-based models, attempt to group messages based on similarities without relying on predefined labels. Reinforcement Learning methods, such as Q-learning and R-learning, take a feedback-driven approach, improving spam detection strategies over time based on rewards or penalties received for correct or incorrect classifications.

For the sake of simplicity and ease of modeling, this project focuses on two widely used supervised learning algorithms: Naive Bayes and Support Vector

Machines (SVM). It's worth noting that modern spam filters often use hybrid models—combinations of multiple algorithms such as Naive Bayes, SVMs, and decision trees. However, this analysis restricts itself to "pure" implementations of Naive Bayes and SVM to allow for clearer modeling and interpretation.

The Naive Bayes classifier is one of the most commonly used models for spam detection. It is based on Bayes' Theorem and assumes that the input features (e.g., words in an email) are independent of each other. Naive Bayes is computationally efficient due to its simplicity and selective use of features [6]. However, this simplicity also limits its ability to capture more complex patterns in spam messages, which can reduce its effectiveness in adversarial settings.

In contrast, the Support Vector Machine (SVM) is a more sophisticated and powerful classifier.. The main idea behind a Support Vector Machine is to transform the data into a new space where it becomes easier to separate the two classes (spam and non-spam messages in this case), and then find the best dividing line or surface that keeps them as far apart as possible. In comparative evaluations, SVM has demonstrated higher classification accuracy, with reported rates around 95%, while Naive Bayes typically achieves an average accuracy of 85.5% [6].

3.2 Payoff Matrix Setup

To model the dynamics of machine learning strategies in spam filtering, we consider a population of spam classifiers, each using either Naive Bayes (NB) or Support Vector Machines (SVM) as their strategy. The fitness of a strategy corresponds to its effectiveness in accurately classifying spam emails—helping users avoid harmful or malicious content.

The interaction between classifiers is modeled using the following payoff matrix:

Strategies	(NB) Naive Bayes	(SVM) Support Vector Machine
(NB) Naive Bayes	(3,3)	(1,5)
(SVM) Support Vector Machine	(5,1)	(4,4)

Table 1: Payoff Matrix for Spam Classifiers

In this payoff matrix, when both players use Naive Bayes, they each receive a payoff of 3, representing a baseline level of effectiveness. When both use Support Vector Machines (SVM), the payoff increases to 4 for each player, reflecting SVM's generally superior accuracy in spam detection. In cases where one player uses Naive Bayes and the other uses SVM, the SVM player receives a payoff of 5 due to its performance advantage, while the Naive Bayes player receives only 1. The use of non-zero payoffs throughout the matrix is intentional, as it reflects the assumption that employing any spam filtering strategy is better than not using one at all.

3.3 Nash Equilibrium Analysis

The strategy pair (SVM, SVM) constitutes a pure strategy Nash Equilibrium. To see this, consider player 1's best response: if player 2 chooses Naive Bayes (NB), player 1 receives a higher payoff by choosing SVM; if player 2 chooses SVM, SVM remains the better response for player 1. The same logic applies to player 2—regardless of whether player 1 plays NB or SVM, player 2 always receives a higher payoff by choosing SVM. Since neither player has an incentive to deviate from SVM when the other is playing it, (SVM, SVM) is a stable equilibrium point.

3.4 Evolutionarily Stable Strategy (ESS) Analysis

To determine whether either strategy in the spam classifier payoff matrix is evolutionarily stable, we examine the conditions for evolutionary stability using the payoffs.

First, we consider a population where everyone plays Naive Bayes (NB). In this case, the expected payoff for NB is 3. If a rare mutant using SVM appears, it earns a payoff of 5 when playing against NB. Since $5 > 3$, SVM can successfully invade a population of NB players, so NB is not evolutionarily stable.

Next, we consider a population where everyone plays SVM. The expected payoff for SVM in this population is 4. A rare mutant using NB earns a payoff of 1 when matched against an SVM player. Since $4 > 1$, SVM performs better against itself than NB does against it, satisfying the first condition of evolutionary stability. Moreover, when matched directly, SVM earns 5 against NB, while NB earns only 1. This satisfies the second condition: that SVM performs better against the mutant than the mutant does against it. Therefore, we can conclude that SVM is an evolutionarily stable strategy.

4 Machine Learning as Spammers

4.1 Overview

Machine learning has become one of the most effective tools for identifying and filtering spam. By training models on large datasets of labeled emails—where each message is marked as either spam or not spam (ham)—these systems learn the patterns and features commonly found in spam. These patterns might include frequent use of certain keywords, suspicious links, or unusual sender behavior. Once trained, the model can apply this knowledge to new emails, helping users avoid unwanted or dangerous content.

However, the same idea can be used from the opposite side. If spam filters rely on machine learning to learn how to detect spam, spammers can also use machine learning to learn how to evade detection. Suppose a spammer has access to a dataset showing which of their emails were marked as spam and which were not. Using this dataset, they can train their own machine learning model to understand the decision boundaries of the spam filter. In other words, they can reverse-engineer the filter's behavior by learning what kinds of emails are likely to pass through and what kinds are likely to be blocked.

This creates a feedback loop where spammers continuously adjust their messages based on how the spam filter responds. For example, they may learn that adding certain "good" words—like "meeting" or "project"—helps an email look more legitimate. They might also avoid using phrases or structures that are common in detected spam. In this way, machine learning becomes a tool not only for defense but also for attack. It highlights the arms race between spam filters and spammers: each side adapting and evolving based on the other's behavior.

4.2 Payoff Matrix Setup

To better understand the choices spammers make when trying to fool a spam filter, we can model their behavior using a payoff matrix. In this simplified game, each spammer has two strategies: to add "good words" to their message (strategy A), or not to add them (strategy D). "Good words" are those that are often seen in legitimate emails—words like "meeting," "project," or "report"—and including them can make spam messages seem more legitimate to the classifier.

Strategies	(A) Add good words	(D) Do not add good words
(A) Add good words	(-1,-1)	(5,0)
(D) Do not add good words	(0,5)	(0,0)

Table 2: Payoff Matrix for Adversarial Attack

As shown in Table 2, if a spammer chooses to add good words while the other does not, the spammer who adds them gains a payoff of 5. This represents the benefit of successfully passing the filter and having their message seen. The other spammer, who does not use this trick, gets 0.

However, if both spammers use the “add good words” strategy, the spam filter will learn from the updated dataset. Over time, the filter becomes better at detecting this tactic, which reduces its effectiveness. In this case, both spammers suffer a penalty, getting a payoff of -1 each. This represents the cost of using a strategy that has now become predictable and easily blocked.

If neither spammer adds good words, they both receive 0—neither gaining nor losing anything. This scenario models a case where the messages are detected as spam or ignored without any clever attempts to bypass the filter.

This simple matrix helps us study how strategies evolve over time when spammers adapt to each other and to changing filters.

4.3 Nash Equilibrium Analysis

To better understand how spammers might behave over time, we analyze the payoff matrix using the concept of Nash Equilibria (NE). A Nash Equilibrium is a set of strategies where no player can improve their outcome by changing their strategy alone. In this context, it helps us figure out which strategies spammers are likely to settle on, assuming each one is trying to maximize their own success.

Pure Strategy Nash Equilibria. We first examine the pure strategies. There are two Nash equilibria in pure strategies: $(\text{Add}, \text{Do not add})$ and $(\text{Do not add}, \text{Add})$. In each of these cases, one spammer benefits by using the “add good words” strategy while the other sticks with the safer option. Neither player would want to change their choice alone, since doing so would reduce their payoff.

The strategy pair (Add, Add) is not a Nash equilibrium. Although both players are using the same strategy, they each receive a negative payoff of -1, which makes them worse off. Either spammer could improve their result by unilaterally switching to “do not add,” which would raise their payoff from -1 to 0.

Similarly, $(\text{Do not add}, \text{Do not add})$ is also not a Nash equilibrium. If either player switches to “add good words” while the other does not, they would increase their payoff from 0 to 5, which is a clear improvement. So at least one player would want to change their strategy in this case.

Mixed Strategy Nash Equilibrium. Besides these pure strategy equilibria, there is also a symmetric mixed-strategy Nash equilibrium. In a mixed strategy, each player randomizes between the two available strategies. Let’s suppose each player chooses “add good words” with probability p and “do not add” with probability $1 - p$.

We can calculate the expected payoff for each strategy:

- For “add good words,” the expected payoff is: $p \cdot (-1) + (1-p) \cdot 5 = 5 - 6p$.
- For “do not add,” the expected payoff is: $p \cdot 0 + (1 - p) \cdot 0 = 0$.

To find the point where both strategies give the same expected payoff (and the player is indifferent between them), we set the two payoffs equal:

$$5 - 6p = 0 \Rightarrow p = \frac{5}{6}.$$

So in the mixed-strategy equilibrium, each spammer chooses to “add good words” with probability $\frac{5}{6}$ and chooses “do not add” with probability $\frac{1}{6}$. This result shows that even when both players are randomizing, they are still heavily favoring the “add good words” strategy most of the time, which reflects the strong short-term benefit it offers—even if it becomes less effective when overused.

4.4 Evolutionarily Stable Strategy (ESS) Analysis

While Nash equilibria help us understand stable outcomes when players act rationally, evolutionarily stable strategies (ESS) add another layer. ESS comes from evolutionary game theory and focuses on what happens when a new or ”mutant” strategy appears in a population that is mostly playing one strategy. An ESS is a strategy that, once common in the population, cannot easily be invaded by a small number of players using a different strategy.

We analyze whether any of the strategies in our spam game are evolutionarily stable.

Pure Strategy: “Add good words”. This strategy is not an ESS. To see why, imagine most spammers are using “Add good words.” When they interact with each other, they each receive a payoff of -1. Now suppose a rare spammer appears who tries the “Do not add” strategy. This mutant gets a payoff of 0 when playing against the majority, which is better than -1. Since the mutant does better, it can invade the population. So, “Add good words” is not stable in the evolutionary sense.

Pure Strategy: “Do not add”. This strategy is also not evolutionarily stable. In a population of spammers all choosing “Do not add,” they each get 0 as a payoff. But now, a rare mutant who chooses “Add good words” will get 5 against these players. That’s a much higher reward, which means the mutant strategy can spread. So “Do not add” is not ESS either.

Mixed Strategy NE. The mixed strategy where players choose “Add” with probability $\frac{5}{6}$ and “Do not add” with probability $\frac{1}{6}$ is a Nash equilibrium, but it does not qualify as an ESS. This is because it is not a *strict* equilibrium: mutants using slightly different probabilities can still get the same expected payoff. Since there's no clear advantage over all other strategies, the mixed NE can be invaded. Therefore, it is not stable in the evolutionary sense.

Conclusion. None of the strategies—neither pure nor mixed—meet the criteria for being evolutionarily stable. This means that the system is likely to keep changing over time, as new strategies can appear and gain an advantage, leading to a constantly shifting balance between how spammers behave. This supports the idea that spam detection is a moving target, and that the arms race between spammers and filters is ongoing.

4.5 Spatial Games

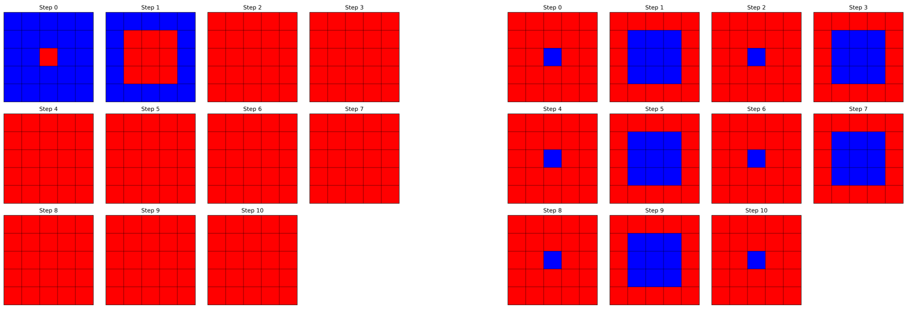
To capture the local interaction dynamics among agents, we extend the analysis to a spatial setting. Agents are placed on a 2D grid and interact only with their neighbors. At each evolutionary step, an agent may update its strategy by imitating the most successful strategy among its local neighborhood, where we use Von Neumann neighborhood for this spatial games simulation, based on accumulated payoffs from interactions.

We implemented a program to initialize the grid and simulate the evolution of strategies over time. The initial configuration and number of steps can be specified, allowing us to explore different invasion dynamics.

Initialization and Simulation Setup We consider two scenarios in the small grid graph setting, with $5 * 5$ and $6 * 6$:

4.5.1 Single Invader in 5×5 Grid

The 5×5 grid simulations reveal distinct patterns depending on the initial invader type. When a single red (Add Good Words) agent occupies the center of an all-blue population, cooperation spreads methodically outward shown in Figure 1a. The initial advantage of the central cooperator becomes apparent by



(a)

(b)

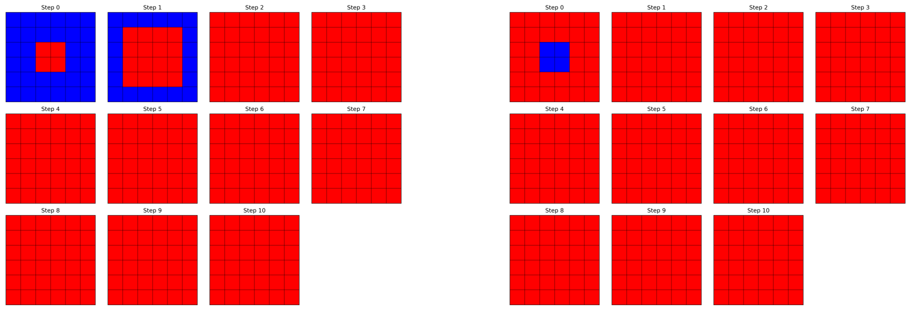
Figure 1: Strategy evolution with single central invader in 5×5 grid. Left: Red invader in blue population. Right: Blue invader in red population.

the second generation, where a stable 3×3 red block emerges. Complete conversion occurs by generation three, demonstrating how localized cooperation can propagate through imitation of successful strategies. This models how spam techniques incorporating legitimate content can become widespread once they demonstrate effectiveness against filters.

The reverse scenario shows markedly different behavior. A single blue defector in an all-red population initially converts its neighborhood, creating a 3×3 blue zone, shown in Figure 1b. However, by generation three, the system reverts to a single central defector, beginning an endless cycle. This oscillation emerges because while defection provides immediate individual benefits, it cannot sustain population-wide dominance against the superior collective payoff of cooperation. In spam terms, this represents techniques that work briefly before filters adapt, forcing constant innovation.

4.6 Four Invaders in 6×6 Grid

The four red invaders in a 6×6 blue grid exhibit faster cooperative takeover than the single invader case, shown in Figure 2a. Multiple nucleation points allow the red strategy to form a 4×4 block by generation two, achieving full dominance by



(a) (b)

Figure 2: Strategy evolution with four central invaders in 6×6 grid. Left: Red invaders in blue population. Right: Blue invaders in red population.

generation four. This accelerated spread models how coordinated adoption of new spam techniques by multiple actors can lead to rapid proliferation through a population.

When we start with four blue players in the middle of a 6×6 grid where everyone else is red, we see some interesting but short-lived changes shown in Figure 2b. In the first round, the blue strategy quickly spreads to take over a big 4×4 area (16 players) as nearby red players switch to blue, tempted by its immediate benefits. But this success doesn't last. By the very next round, something surprising happens - every player suddenly switches back to red. While the blue area grows fast at first, the players at its edges become easy targets for the remaining red players at the grid's borders. There's no stable middle ground where blue can survive - it either takes over completely or disappears entirely. This quick flip from mostly blue back to all red is very different from what we see in smaller grids, where the back-and-forth between strategies continues longer. The larger grid size makes these strategy changes happen more suddenly.

5 Evolutionary Dynamics Between Spam Classifiers and Spammers

To model the interaction between spam filters (defenders) and spammers (attackers), we set up an asymmetric game, where each population has different strategy sets and payoffs. The spam classifiers, representing the defender population, can choose between two strategies: using a Naive Bayes (NB) classifier or a Support Vector Machine (SVM) classifier. On the other hand, the spammers, representing the attacker population, also have two strategic options. They can either employ the ”Add Good Words” (A) strategy or opt for the ”Don't Add Good Words” (D) strategy. This is represented by the following payoff matrix:

Strategies	(A) Add good words	(D) Do not add good words
(NB) Naive Bayes	(2,3)	(4,1)
(SVM) Support Vector Machine	(4,1)	(5,0)

Table 3: Payoff Matrix for Spam Classifiers vs Spammers

The values in the payoff matrix are based on qualitative comparisons of the relative strengths of different strategies in spam classification and evasion. Below are the key assumptions that inform the assignment of these values:

- **Classifier Performance:** Support Vector Machines (SVM) are assumed to be more accurate than Naive Bayes (NB) in detecting spam, particularly when the spam content has been manipulated. This is reflected in SVM generally receiving higher payoffs than NB across all matchups.

- **Spammer Effectiveness:** Spammers are more successful when classifiers are weaker or when they use evasion techniques such as adding legitimate ”good” words to spam messages. Thus, the spammer's payoff is higher when using the ”Add Good Words” (A) strategy against NB, and lower against SVM.

- **Relative Difficulty:** The "Add Good Words" strategy makes spam harder to detect, particularly for simpler models like NB, so NB receives a lower payoff (2) while the spammer gains a higher one (3). Against SVM, (A) is less effective, so SVM retains a high payoff (4), and the spammer's payoff drops to 1.

- **Baseline Effectiveness:** Even NB is assumed to perform reasonably well against standard spam with no added good words (D), earning it a payoff of 4. SVM, being more robust, earns a maximum payoff of 5 in this setting, while the spammer earns nothing (0), representing complete detection.

- **Non-zero Payoffs:** All classifier payoffs are kept above zero to reflect the assumption that deploying any spam filter—regardless of type—is better than having none.

Nash Equilibrium Analysis:

To identify pure strategy Nash equilibria, we look for strategy pairs where neither player has an incentive to deviate unilaterally.

- Fixing the spammer's strategy:
  - If the spammer plays A:
    - Classifier payoff: NB = 2, SVM = 4 $\Rightarrow$ Classifier prefers SVM.
  - If the spammer plays D:
    - Classifier payoff: NB = 4, SVM = 5 $\Rightarrow$ Classifier prefers SVM.

  So the best response for the classifier is always SVM.

- Fixing the classifier's strategy:
  - If the classifier plays NB:

    - Spammer payoff: A = 3, D = 1 ⇒ Spammer prefers A.
  - If the classifier plays SVM:
    - Spammer payoff: A = 1, D = 0 ⇒ Spammer prefers A.

So the best response for the spammer is also always A.

Conclusion: The strategy pair (SVM, A) is a pure strategy Nash equilibrium, as both players are playing their best responses.

In modeling this interaction, the payoff matrix reflects the asymmetry in goals: classifiers aim to maximize accuracy, while spammers seek to evade detection. Our analysis shows that SVM consistently outperforms Naive Bayes against both types of spammer strategies, providing a higher payoff due to its superior classification accuracy. For spammers, adding good words proves more effective against both classifiers, even though it incurs a cost. As a result, the unique Nash equilibrium occurs when classifiers always use SVM and spammers always add good words. This strategy pair is also evolutionarily stable, meaning that neither population has an incentive to deviate when most of the population adopts these strategies. Over time, this leads to a stable but adversarial equilibrium where classifiers evolve toward more complex models like SVM, and spammers continuously adapt their tactics to exploit statistical weaknesses—illustrating the ongoing co-evolution between detection systems and adversarial behavior in real-world spam filtering.

6 Conclusion

Our game-theoretic analysis of spam filtering reveals several key insights. First, the interaction between spammers and filters naturally leads to an arms race, with neither side able to establish a stable, long-term advantage. The Nash equilibrium and ESS analyses show that while certain strategies may dominate temporarily, they remain vulnerable to invasion by new approaches. This explains why spam techniques continually evolve and why filters must constantly adapt.

The spatial game simulations further demonstrate how local interactions shape global outcomes. Cooperative strategies (adding good words) can spread effectively from small clusters, suggesting that coordinated efforts among spammers may lead to widespread adoption of new evasion techniques. Conversely, defecting strategies create unstable patterns, alternating between rapid expansion and sudden collapse. This mirrors real-world observations where new spam methods gain temporary popularity before filters adapt.

These findings have practical implications for spam filter design. Static detection systems will inevitably fail against evolving spam tactics. Instead, filters should incorporate: (1) continuous learning from new data, (2) mechanisms to detect and respond to strategy shifts among spammers, and (3) localized adaptation to address regional variations in spam techniques. Future work could explore more complex network structures and real-world datasets to refine these models.

However, it is important to note the limitations of this analysis. In this work, we made several assumptions to simplify the problem and adapt machine learning models into a game-theoretic framework. For instance, we assumed fixed payoff values and strategic choices for both spam classifiers and spammers, without fully accounting for the complexities of real-world scenarios, such as changing network dynamics, evolving user behavior, or varying levels of attack sophistication. The use of specific numerical values in the payoff matrix, while necessary for model simplicity, oversimplifies the nuanced and dynamic nature of the interaction between spam filters and spammers. Additionally, our analysis assumes a binary classification model and does not capture the full range of potential filtering algorithms and evolving spam strategies. Future work should consider relaxing these assumptions and incorporate more diverse, realistic scenarios to better capture the intricacies of the real-world arms race between spammers and spam filters.

Ultimately, our analysis shows that spam filtering is not just a classification problem but an evolutionary struggle. By understanding the game-theoretic dynamics at play, we can develop more robust, adaptive systems that keep pace with spammers' ever-changing strategies.

References

[1] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] B. Biggio and F. Roli, “Wild patterns: Ten years after the rise of adversarial machine learning,” *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[4] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam filtering with naive bayes–which naive bayes?,” in *CEAS*, Citeseer, 2006.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, 2015.

[6] A. R. A. H. K. D. A. B. S. T. Ahmed, Naeem, “Machine learning techniques for spam detection in email and iot platforms: Analysis and research challenges,” *Security and Communication Networks*, 2022.