Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Characterising Mobile Money Phishing Attacks Using Imitation Learning Relying on Expert Demonstrations

Alima Nzeket Njoya
Cybersecurity with Computational and Artificial Intelligence Research Group
Claudette Nyinawumuntu
African Institute for Mathematical Sciences (AIMS)
Franklin Tchakounté
Cybersecurity with Computational and Artificial Intelligence Research Group
Derrick Méthode Bagaza
dmbagaza@univ-bangui.org

University of Bangui
Mismadji Sosthène Togbé
Cybersecurity with Computational and Artificial Intelligence Research Group

Research Article

Keywords: Mobile money, Phishing, Imitation learning, Expert demonstrations, Behavior cloning, Generative adversarial imitation learning

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11276572/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Characterising Mobile Money Phishing Attacks Using Imitation Learning Relying on Expert Demonstrations

Alima Nzeket Njoya[1,2*], Claudette Nyinawumuntu[3], Franklin Tchakounte[1,2], Derrick Méthode Bagaza[4*], Mismadji Sosthène Togbé[1,2,5]

[1]Cybersecurity with Computational and Artificial Intelligence Research Group, Ngaoundéré, Cameroon.
[2]Department of Mathematics and Computer Science, Faculty of Science, University of Ngaoundéré, Ngaoundéré, Cameroon.
[3]African Institute for Mathematical Sciences (AIMS), Kigali, Rwanda.
[4*]Department of Computer Science, Higher Institute of Technology, University of Bangui, Bangui, Central African Republic.
[5]African Institute for Mathematical Sciences (AIMS), Limbe, Cameroon.

*Corresponding author(s). E-mail(s): dmbagaza@univ-bangui.org; alimanzeket@gmail.com;
Contributing authors: claudette.nyinawumuntu@aims.ac.rw; franklin.tchakounte@univ-ndere.cm; mismadji.togbe@aims-cameroon.org;

Abstract

Mobile money has become hugely popular in Africa, but its growth has been shadowed by phishing, a form of social engineering in which fraudulent messages, emails, or calls trick users into revealing sensitive information and surrendering their funds. Designing effective countermeasures first requires understanding how attackers actually operate. This paper characterises mobile money phishing with imitation learning relying on expert demonstrations: instead of hand-crafting a reward, as prior reinforcement learning approaches must, at the cost of sparse rewards and limited scalability, we learn the attacker's strategy directly from examples of the attacker's own behaviour. Treating the attacker as the expert, we formalise the attack as a four-state process in which transitions depend jointly on the attacker's actions and on the victim's reactions, build a simulated dataset

of expert demonstrations (state–action pairs) from the attack scenario observed in real cases, and apply two imitation learning algorithms to it: Behavior Cloning (BC) and Generative Adversarial Imitation Learning (GAIL). Behavior Cloning reproduced the attacker's actions accurately, performing best at a learning rate of 0.006, though it struggled with the branching decisions at the exit state. In GAIL, the learned policy generated state–action pairs while a discriminator separated them from those of the attacker's demonstrations; across training the discriminator loss decreased steadily and the generator loss rose, showing that the learned policy remained distinguishable from the attacker within the training budget. The results indicate that imitation learning from expert demonstrations offers a promising lens for understanding and ultimately countering mobile money phishing.

Keywords: Mobile money, Phishing, Imitation learning, Expert demonstrations, Behavior cloning, Generative adversarial imitation learning

1 Introduction

Mobile money lets people deposit, withdraw, send, and receive funds, and reach a range of financial services, directly from a mobile phone [1]. The model has spread worldwide, but nowhere has it taken hold as firmly as in Africa. In 2021, the volume of mobile money transactions on the continent rose by 39% to USD 701.4 billion, with Africa accounting for roughly 70% of the global market; globally, the total value of transactions reached USD 1.045 trillion, a 31% increase over 2020 that the GSMA attributed in part to the COVID-19 pandemic [2]. By reducing reliance on cash and traditional banking, mobile money has proved especially valuable for the previously unbanked. Rwanda illustrates the trend. MTN launched the first service there in 2010, followed by Tigo in 2011 and Airtel in 2013 [3]. By 2021, MTN had registered 6.2 million subscribers and Airtel around 5 million [4], and during the pandemic MTN Rwanda reported a 450% jump in mobile money transactions over three months, from 7.2 to 40 billion Rwandan francs [5].

This convenience carries a cost. Social engineering, the manipulation of people into revealing private information, and its best-known variant, phishing, exploit human trust rather than technical flaws [6]. They may be unsophisticated, but they are damaging, and untrained users remain the weakest link [7]. Attackers pose as banks or telecom operators and request personal information by call, email, or text [8, 9]. The consequences for mobile money are tangible: in Rwanda in 2020, fraudulent activity was blamed for an estimated loss of 12 million Rwandan francs, with 80 complaints filed in just two months during the pandemic [10].

Most prior work treats this as a detection problem. In our earlier work, Nzeket Njoya et al. [11] applied reinforcement learning (RL) to characterise and detect mobile money phishing, but the approach inherited the familiar weaknesses of RL: difficulty in computing truly optimal policies, limited scalability of value-function representations, reliance on trial-and-error that produces sub-optimal mistakes, and no guarantee of identifying the best path upfront. The present paper extends that study. We argue

that to reproduce phishing behaviour convincingly, one should instead learn directly from the attacker's own actions. We therefore propose an imitation learning framework and make the following contributions:

- We formalise mobile money phishing as an imitation learning problem, modelling the attacker–victim interaction as a four-state process in which the attacker's actions, the victim's reactions and the resulting transition probabilities are defined explicitly (Section 3).
- We apply two complementary algorithms, Behavior Cloning (BC) and Generative Adversarial Imitation Learning (GAIL), to attacker demonstrations, and analyse what each reveals about attacker strategy (Sections 3 and 4).
- We analyse the GAIL training dynamics and clarify that they measure how closely the learned policy imitates the attacker (Section 4).

The remainder of the paper reviews the relevant background and related work (Section 2), presents the methodology (Section 3), reports and discusses the results (Section 4), and concludes with limitations and future directions (Section 5).

2 Background and Related Work

2.1 Social Engineering and Phishing

A social engineering attack manipulates people into divulging confidential information by exploiting emotions such as trust, fear, greed, and curiosity [12]. It typically proceeds in four stages: gathering information about the target; building a relationship and earning trust; exploiting that trust to extract information and launch the attack; and exiting without leaving a trace [12–14]. Phishing is the social engineering variant that uses deceptive emails, messages, calls, or websites to trick victims into surrendering credentials or personal data [15, 16]. For mobile money, the dominant vectors are smishing (SMS phishing), in which a text appears to come from a legitimate operator or bank and prompts the victim to follow a link or call a number, and vishing (voice phishing), in which the attacker calls while posing as a trusted agent [17, 18]. Our own examination of cases in Rwanda involving MTN Rwanda and Airtel-Tigo, drawing on customer complaints, operator awareness campaigns, and direct observation, confirmed that smishing and vishing are the most common.

2.2 From Reinforcement to Imitation Learning

In reinforcement learning, an agent in state s_t takes an action a_t according to a policy π, receives a reward r_t, and transitions to a new state, aiming to maximise its cumulative reward [19, 20]. A policy maps a state to a distribution over actions,

$$\pi(s) = \Pr(A \mid s), \quad s \in \mathcal{S}, \tag{1}$$

and the optimal policy maximises the expected return,

$$\pi^* = \arg\max_{\pi} \mathbb{E}(R \mid \pi). \tag{2}$$

RL falters, however, when rewards are sparse or when a reward function is hard to design by hand, both of which apply to modelling a human attacker [11]. Imitation learning (IL) sidesteps this. Instead of a reward signal, it uses expert demonstrations: the learner observes state–action pairs produced by an expert and recovers a policy that reproduces them [21, 22]. The goal is to learn a policy π that approximates the expert policy π_E, such that $\pi(s_i) \approx \pi_E(s_i) = a_i$. IL has been applied successfully to robotics, autonomous driving, and game playing, settings where demonstrations are easier to obtain than well-shaped rewards. We use two IL algorithms. Behavior Cloning casts imitation as supervised learning: given a dataset $\mathcal{D} = \{(s_1, a_1), \ldots, (s_n, a_n)\}$ of expert state–action pairs, it learns a policy by minimising a loss between predicted and expert actions [23, 24],

$$\hat{\pi} = \arg\min_{\pi} \sum_{(s_i, a_i) \in \mathcal{D}} \mathcal{L}\big(\pi(s_i), a_i\big). \tag{3}$$

GAIL [25] borrows from generative adversarial networks: the policy being learned acts as a generator of state–action pairs, while a discriminator is trained to distinguish the pairs of the expert demonstrations from those produced by the policy. The two train adversarially, the policy trying to fool the discriminator and the discriminator trying to classify correctly, with the discriminator output serving as a learned reward signal [26].

2.3 Related Work on Mobile Money Fraud

A range of approaches has been proposed to control mobile money fraud. Akomea-Frimpong et al. [27] traced high fraud rates in Ghana to weak controls and inadequate training, recommending legal codes and awareness campaigns. Several studies frame fraud as a supervised classification task: Mohammed [28] found random forest best among classical and deep models for SMS fraud; Botchey et al. [29] favoured gradient-boosted trees but reported imbalanced data and overfitting; and Lokanan [30] again found random forest strongest for transaction fraud, while cautioning about biased data. El Ayeb et al. [31] applied community-detection algorithms on simulated data, limited by class imbalance, and Chebii [32] built SAFE-CASH, a prototype focused narrowly on smishing and vishing. Closest to our work, Nzeket Njoya et al. [11] characterised mobile money phishing with reinforcement learning (Q-learning, Markov decision processes and deep RL), but encountered the scalability and optimality limits noted above.

Two recent studies published in this journal illustrate the current state of learning-based phishing defence. Khan [33] proposed HARE, a deep reinforcement learning detector whose reward is shaped by scalar ratings from trained human experts; it reached an F1-score of 97.62% on the PhiUSIIL URL dataset, but it still depends on a reward signal that must be designed and calibrated by human raters. Sharafat et al. [34] combined convolutional, bidirectional recurrent and attention layers to classify spam and phishing emails, with in-domain accuracies above 98% on four benchmark corpora; under cross-dataset evaluation, however, accuracy fell to 61.10% on Enron, a drop the authors attribute to linguistic and distributional shift between corpora. Both

works treat phishing as a classification problem on the defender's side and learn from the content of messages or URLs.

Table 1 summarises these works. Our contribution differs on two counts. First, instead of classifying messages, we model the attacker's strategy across the successive stages of the attack, which content-based detectors such as [34] do not capture. Second, imitation learning learns this strategy directly from demonstrations, without the hand-designed or human-shaped reward that reinforcement learning approaches require [11, 33].

Table 1 Selected approaches to controlling phishing on mobile money services, and recent learning-based phishing detectors published in this journal.

Year	Author	Approach	Main limitation
2019	Akomea-Frimpong et al. [27]	Legal code, internal fraud policy	No dataset
2019	Mohammed [28]	ML/DL (RF, SVM, LSTM, CNN)	Limited dataset
2020	Botchey et al. [29]	Supervised ML (SVM, GBDT, NB)	Imbalance, overfitting
2020	El Ayeb et al. [31]	Community detection	Unbalanced classes
2021	Chebii [32]	SAFE-CASH prototype	Narrow scope
2022	Lokanan [30]	ML (LR, RF)	Biased dataset
2023	Nzeket Njoya et al. [11]	Reinforcement learning	Scalability, optimality
2025	Khan [33]	RL with human feedback	Reward to design
2026	Sharafat et al. [34]	CNN–BiGRU–attention	Cross-dataset drop

BiGRU, bidirectional gated recurrent unit; CNN, convolutional neural network; DL, deep learning; GBDT, gradient-boosted decision trees; LR, logistic regression; LSTM, long short-term memory; ML, machine learning; NB, naive Bayes; RF, random forest; RL, reinforcement learning; SVM, support vector machine.

3 Methodology

3.1 Formalising the Attack

We cast mobile money phishing as an imitation learning problem in which the expert is the attacker, the learner is a policy trained to reproduce the attacker's behaviour, and the environment comprises the states of the attack and the victim's reactions. Drawing on the second through fourth stages of a social engineering attack, we define four states (Table 2):

1. **Initial state (S_1).** The attacker has gathered information about the victim and is planning the approach, by call or text, and starting to build a relationship.
2. **Vulnerability state (S_2).** The pivotal state, where the attacker reads the victim's emotions and tries to instil confidence. Success advances the attack; suspicion sends the attacker back to S_1.
3. **Attack state (S_3).** The attacker waits for the victim to act, by sending money or information. The victim may still detect the scam, forcing the attacker to stay or retreat.

4. **Data collection and exit (S_4).** The attacker cuts contact to avoid being traced. If the goal has been obtained, the attacker seeks the next target; otherwise it either abandons the victim or tries again.

Table 2 States of a mobile money phishing attack and their associated actions.

Symbol	State	Representative actions
S_1	Initial state	Collect victim information; send fake message; send email
S_2	Vulnerability state	Call to read emotions and build trust; converse via social media
S_3	Attack state	Transfer money to attacker; obtain password/information
S_4	Data collection & exit	Recover balance; exit and cease contact

3.2 Attacker-Victim Interaction Model

We now make the interaction between the two parties explicit. Let $\mathcal{S} = \{S_1, S_2, S_3, S_4\}$ be the states of Table 2, to which we add two terminal outcomes: $S_\top$, success (the attacker has obtained money or credentials and leaves), and $S_\perp$, failure (the attacker gives up on the victim).

Attacker actions. At each step the attacker chooses a move

$$a \in \mathcal{A} = \{\mathsf{Stay}, \mathsf{Up}, \mathsf{Down}, \mathsf{Left}, \mathsf{Right}\}. \tag{4}$$

The states are laid out on a 2×2 grid: S_1 at the bottom left, S_2 above it, S_3 to the right of S_2 and S_4 below S_3 (Fig. 1). Each move therefore corresponds to a concrete decision in the phishing scenario, given in Table 3. Up in S_1 and Right in S_2 are the *advancing* moves: the attacker asks something of the victim (to answer the call or message, then to make the transfer or give the PIN), and the move succeeds only if the victim complies. Down in S_2 and Left in S_3 are voluntary retreats, and Down in S_3 means that the attacker collects what it has obtained and cuts contact. A move that would leave the grid has no effect and is equivalent to Stay.

Victim reactions. After each move, the victim reacts with

$$r \in \mathcal{R} = \{+, 0, -\}, \tag{5}$$

where $+$ is a compliant reaction (the victim answers, trusts, pays or discloses), 0 a neutral one (hesitation, no answer, request for details) and $-$ a negative one (suspicion, refusal, blocked number). The victim's strategy is the conditional distribution

$$\rho(r \mid s, a), \qquad \sum_{r \in \mathcal{R}} \rho(r \mid s, a) = 1, \tag{6}$$

so that $\rho(- \mid s, a)$ is the probability that the victim detects the scam when the attacker plays a in state s.

Attacker strategy. The attacker adapts its next move to the way the victim has just reacted. We summarise this information by $c_t = \mathbf{1}[r_{t-1} = +]$, with $c_0 = 0$, and write the attacker's strategy, i.e. the expert policy, as

$$a_t \sim \pi_E(\cdot \mid s_t, c_t). \tag{7}$$

The two values of c give the two scenarios reported in Section 4.1: in scenario (a), $c = 0$, the victim's last reaction was neutral or negative; in scenario (b), $c = 1$, the victim has just complied.

Transition conditions. A flag $g \in \{0, 1\}$ records whether the attacker has obtained its goal. Given (s_t, a_t, r_t), the next state is $s_{t+1} = f(s_t, a_t, r_t)$, with

$$f(s, a, r) = \begin{cases} S_4 \text{ and } g = \mathbf{1}[r = +] & \text{if } s = S_3 \text{ and } a = \mathsf{Down}, \\ \mathrm{prev}(s) & \text{else if } s \in \{S_2, S_3\} \text{ and } r = -, \\ \mathrm{next}(s) & \text{else if } a \text{ is advancing and } r = +, \\ m(s, a) & \text{otherwise}, \end{cases} \tag{8}$$

where $\mathrm{next}(S_1) = S_2$, $\mathrm{next}(S_2) = S_3$, $\mathrm{prev}(S_2) = S_1$, $\mathrm{prev}(S_3) = S_2$, and $m(s, a)$ is the state reached by a move that does not depend on the victim's consent: S_1 for Down in S_2, S_2 for Left in S_3, and s itself for Stay, for a move that leaves the grid, or for an advancing move that the victim did not accept. In other words, a positive reaction lets an advancing move succeed, a neutral reaction keeps the attacker where it is, and a negative reaction in S_2 or S_3 sends it back one state, as described in Section 3.1.

Terminal outcomes. If S_4 is reached with $g = 1$, the episode ends in success, $S_\top$. If it is reached with $g = 0$, the victim refused or did not complete the transfer when the attacker tried to collect. The attacker then has three options: Left, restart the approach with a new pretext (S_1); Up, contact the victim again to rebuild trust (S_2); any other move, abandon the victim ($S_\perp$). An episode that has not reached $S_\top$ after $T_{\max}$ steps is also counted as a failure.

Table 3 Meaning of each move in each state. "No effect" denotes a move that would leave the grid; it is equivalent to Stay. In S_2 and S_3, a negative reaction of the victim sends the attacker back one state whatever the move, except Down in S_3 (Eq. (8)).

Move	In S_1	In S_2	In S_3	In S_4 ($g = 0$)
Stay	Keep gathering information or resend the bait	Keep the conversation going to build trust	Wait for the victim to transfer or disclose	Abandon the victim $\to S_\perp$
Up	Contact the victim (call, SMS) $\to S_2$ if $r = +$	No effect	No effect	Contact the victim again $\to S_2$
Right	No effect (S_1 and S_4 are not connected)	Ask for the transfer or the PIN $\to S_3$ if $r = +$	No effect	Abandon the victim $\to S_\perp$
Down	No effect	Withdraw and restart the approach $\to S_1$	Collect and cut contact $\to S_4$	Abandon the victim $\to S_\perp$
Left	No effect	No effect	Back off to rebuild trust $\to S_2$	New pretext $\to S_1$

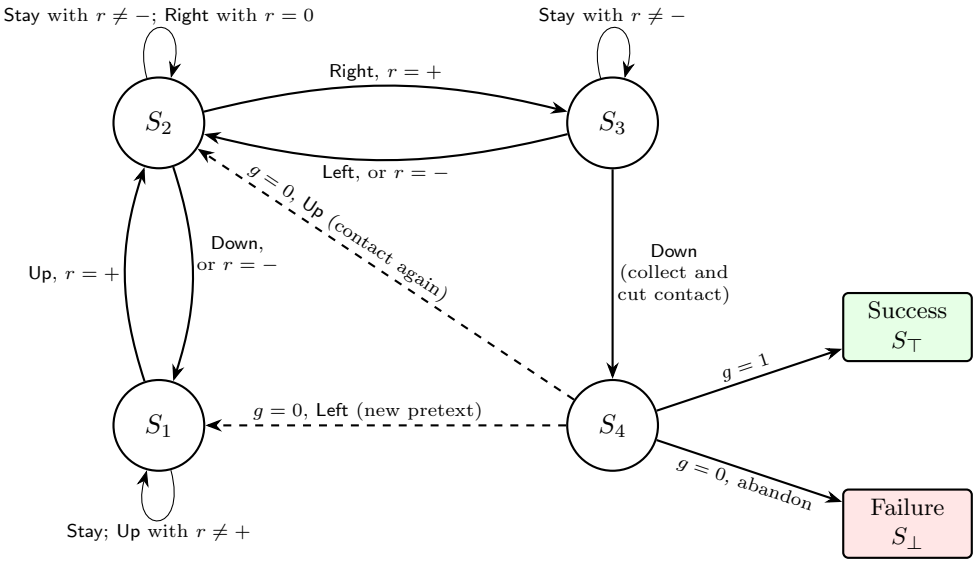


Fig. 1 Transition diagram of the attacker–victim interaction. Each edge is labelled with the attacker's move and, when relevant, the victim's reaction that triggers it. Dashed edges are the options left to the attacker after an exit without gain ($g = 0$). Moves that would leave the grid are treated as Stay.

Transition probabilities. Combining both strategies, the probability that the attack moves from state s to state s' is

$$P(s' \mid s, c) = \sum_{a \in \mathcal{A}} \pi_E(a \mid s, c) \sum_{r \in \mathcal{R}} \rho(r \mid s, a)\, \mathbf{1}\big[f(s, a, r) = s'\big]. \tag{9}$$

For the main transitions of the attack, Eqs. (8) and (9) give

$$P(S_2 \mid S_1, c) = \pi_E(\mathsf{Up} \mid S_1, c)\, \rho(+ \mid S_1, \mathsf{Up}), \tag{10}$$

$$P(S_3 \mid S_2, c) = \pi_E(\mathsf{Right} \mid S_2, c)\, \rho(+ \mid S_2, \mathsf{Right}), \tag{11}$$

$$P(S_1 \mid S_2, c) = \pi_E(\mathsf{Down} \mid S_2, c) + \sum_{a \neq \mathsf{Down}} \pi_E(a \mid S_2, c)\, \rho(- \mid S_2, a), \tag{12}$$

$$P(S_2 \mid S_3, c) = \pi_E(\mathsf{Left} \mid S_3, c) + \sum_{a \notin \{\mathsf{Left}, \mathsf{Down}\}} \pi_E(a \mid S_3, c)\, \rho(- \mid S_3, a), \tag{13}$$

$$P(S_\top \mid S_3, c) = \pi_E(\mathsf{Down} \mid S_3, c)\, \rho(+ \mid S_3, \mathsf{Down}). \tag{14}$$

The attacker controls π_E and the victim controls ρ: an attack progresses only when the attacker chooses an advancing move and the victim complies, and it regresses either by the attacker's choice or by the victim's detection. The probability of a complete attack trajectory $\tau = (s_0, c_0, a_0, r_0, \ldots, s_T)$ is

$$P(\tau) = \prod_{t=0}^{T-1} \pi_E(a_t \mid s_t, c_t)\, \rho(r_t \mid s_t, a_t)\, \mathbf{1}\big[s_{t+1} = f(s_t, a_t, r_t)\big], \tag{15}$$

and the success rate of the attacker against a victim of strategy ρ is the probability that τ reaches $S_\top$. Imitation learning does not require this quantity, or any reward, to be specified. It only needs demonstrations $\mathcal{D} = \{(x_i, a_i)\}_{i=1}^{n}$, with $x_i = (s_i, c_i)$, produced by the attacker interacting with the victim, from which it estimates a policy $\pi_\theta(a \mid x) \approx \pi_E(a \mid x)$.

3.3 Behavior Cloning

Behavior Cloning trains a neural-network policy to imitate the attacker from the demonstration dataset (Algorithm 1). Over a number of epochs, it feeds the observations $x = (s, c)$ to the network, compares predicted actions against the expert's via cross-entropy loss, and updates the parameters by gradient descent.

Algorithm 1 Behavior Cloning

Require: Expert dataset $\mathcal{D} = \{(x_1, a_1), \ldots, (x_n, a_n)\}$; learning rate α; epochs N; batch size B

Ensure: Policy π_θ that mimics expert behaviour

Initialise neural-network policy π_θ and optimiser with rate α
for $i \leftarrow 1$ to N **do**
  for each mini-batch $\mathcal{B}$ of size B **do**
    Compute predicted action distribution $\hat{y} = \pi_\theta(\cdot \mid x)$
    Compute loss $\mathcal{L} = -\sum_{(x,a)\in\mathcal{B}} \log \hat{y}(a \mid x)$
    Update $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$
  end for
end for
return learned policy π_θ

3.4 Generative Adversarial Imitation Learning

GAIL (Algorithm 2) learns a policy π_θ that reproduces the expert's behaviour. A discriminator $D_\phi(x, a)$ outputs the probability that a state–action pair comes from the expert demonstrations rather than from π_θ, and the two are trained on

$$\min_\theta \max_\phi \; \mathbb{E}_{\pi_E}\big[\log D_\phi(x,a)\big] + \mathbb{E}_{\pi_\theta}\big[\log\big(1 - D_\phi(x,a)\big)\big] - \lambda\, \mathcal{H}(\pi_\theta), \tag{16}$$

where $\mathcal{H}(\pi_\theta)$ is the policy entropy, which encourages exploration. The discriminator output gives the reward used to update the policy, which plays the role of the generator. In our setting, the expert pairs are the attacker's demonstrations and the generated pairs are the policy's imitation of them. The discriminator therefore measures how far the learned policy is from the attacker; it is not trained to separate legitimate from fraudulent behaviour.

Algorithm 2 Generative Adversarial Imitation Learning (GAIL)

Require: Expert demonstrations $\mathcal{D}$; discriminator D_ϕ; policy (generator) π_θ; rate α; entropy weight λ; iterations T

Ensure: Policy π_θ

Initialise ϕ and θ randomly
for $t \leftarrow 1$ to T **do**
  Sample expert batch $\mathcal{D}_{\text{batch}}$ from $\mathcal{D}$
  Generate pairs (x_i, a_i) by running the current policy π_θ
  Update ϕ to maximise $\sum_{(x,a)\in\mathcal{D}_{\text{batch}}} \log D_\phi(x,a) + \sum_i \log\big(1 - D_\phi(x_i, a_i)\big)$
  Compute rewards $r_i = \log D_\phi(x_i, a_i)$
  Update θ to maximise $\sum_i r_i + \lambda\, \mathcal{H}(\pi_\theta)$
end for
return policy π_θ

3.5 Experimental Setup

We built a simulated dataset of expert demonstrations, following the attack scenario derived from the cases described in Section 2.1; it captures the attacker's actions together with the corresponding states and the victim's reactions. Observations were encoded as feature vectors and actions as one-hot vectors; the data was shuffled and split into training and test sets. For BC, we trained a neural network with hidden layers using ReLU and softmax activations, optimised with cross-entropy loss and the Adam optimiser. For GAIL, we trained the policy and the discriminator iteratively. We evaluated both with accuracy and loss, and BC additionally with confusion matrices.

4 Results and Discussion

4.1 Behavior Cloning

Tuning the hyper-parameters showed that a learning rate of 0.006 gave the best accuracy and smooth convergence (Fig. 2). A higher rate of 0.01 caused overfitting, the model specializing on the training set, and generalizing poorly, while 0.0001 was too slow to learn. The accuracy and loss curves at the chosen rate (Fig. 3) show a close tracking of training and validation, with accuracy approaching 1.0 and loss approaching 0.

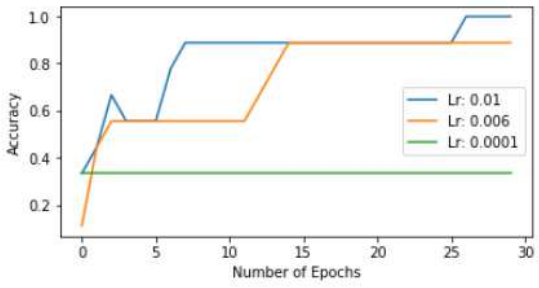


Fig. 2 Behavior Cloning accuracy across epochs for three learning rates. The rate 0.006 converges smoothly to high accuracy; 0.01 overfits and 0.0001 underfits

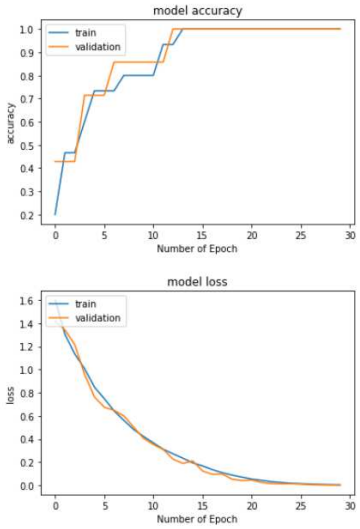


Fig. 3 Model accuracy (top) and loss (bottom) on the training and validation sets at learning rate 0.006

The confusion matrices confirm that the learned policy recognizes the attacker's actions well, both on the training set (Fig. 4) and on the unseen test set (Fig. 5). The persistent difficulty arose at S_4: an attacker may reach the exit state without achieving the goal, then decide whether to return to S_2 or S_1 or abandon the victim (Table 3), a branching choice that is difficult to predict and exposes the limits of a purely supervised approach.

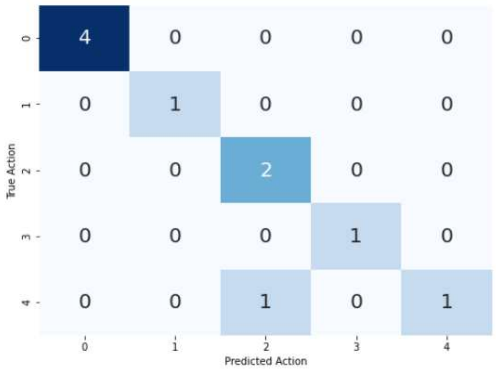


Fig. 4 Confusion matrix on the training set; predictions are largely diagonal

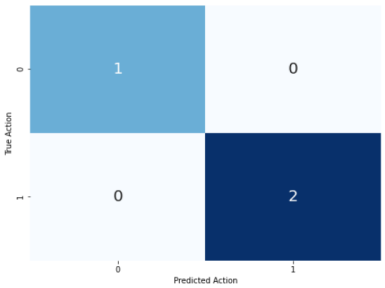


Fig. 5 Confusion matrix on the test set; all test actions are correctly predicted

We also estimated the action probabilities $\hat{\pi}(a \mid s, c)$ of the learned policy (Table 4); the corresponding transition probabilities follow from Eqs. (9)–(14) once the victim's reactions are taken into account. From S_1, the attacker moves up towards S_2 with high probability (0.945): early on, the victim, unaware of the deception, tends to react positively, so the contact is usually accepted. State S_2 is the turning point. In scenario (a) the attacker most often stays (0.596), continuing to interact and draw out more information, and otherwise withdraws to S_1 (0.236); in scenario (b), after a positive reaction, the attacker moves right, i.e. asks for the transfer, with probability 0.726. A similar pattern holds at S_3, where the attacker either holds position (0.541 in scenario (a)) or moves down towards the exit (0.613 in scenario (b)). The probability given to moves that would leave the grid, such as Up in S_2, has the same effect as Stay. The attacker's behaviour is thus governed by the victim's reactions: neutral responses prolong the current state, negative ones force a retreat, and positive ones enable progress towards the goal.

Table 4 Estimated action probabilities $\hat{\pi}(a \mid s, c)$ of the learned policy in each state. For S_2 and S_3, the two rows correspond to scenario (a), where the victim's last reaction was not positive ($c = 0$), and scenario (b), where the victim has just complied ($c = 1$). Each row sums to one.

State	Scenario	Stay	Up	Left	Right	Down
S_1	–	0.0135	0.9450	0.0029	0.0386	$< 10^{-4}$
S_2	(a) $c = 0$	0.5961	0.0811	0.0187	0.0680	0.2361
	(b) $c = 1$	0.0852	0.1793	0.0005	0.7260	0.0090
S_3	(a) $c = 0$	0.5409	0.0026	0.0644	0.0032	0.3889
	(b) $c = 1$	0.2792	0.0156	0.0023	0.0895	0.6134

4.2 Generative Adversarial Imitation Learning

In GAIL, the generator is the policy being learned: it produces state–action pairs meant to resemble the attacker's. The discriminator receives either pairs from the attacker's demonstrations or pairs produced by the policy, and must tell which is which. Both sources therefore describe attacker behaviour, and neither of them represents the victim.

Across 50 training epochs, the discriminator loss fell steadily, from about 1.35 to about 0.5, while the generator loss rose from about 0.75 to about 1.4 (Fig. 6). At the start, the discriminator loss is close to $2 \ln 2 \approx 1.39$, the discriminator loss obtained from Eq. (16) when D_ϕ outputs $1/2$ for every pair, that is, when it cannot tell the two sources apart. Its decrease, together with the rise of the generator loss, shows that the discriminator separated the generated pairs from the demonstrations more and more easily. Within 50 epochs the policy therefore did not reach the attacker's state–action distribution, and the discriminator dominated the adversarial game, a known balance problem of GAIL. These curves describe the quality of the imitation. Since no legitimate behaviour enters the comparison, they cannot be read as a victim becoming better at detecting phishing.

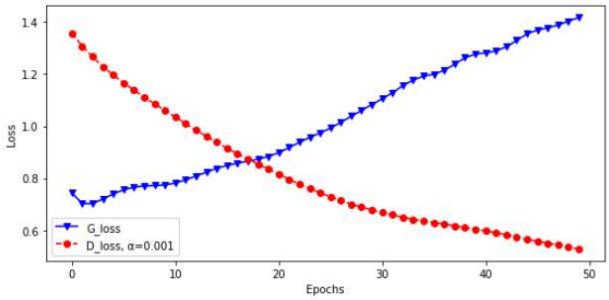


Fig. 6 Generator loss (rising) and discriminator loss (decreasing) over 50 training epochs in GAIL, with a discriminator learning rate of 0.001

We further examined how the reward responds to the learning rate (Fig. 7). In GAIL the reward is derived from the discriminator's output, read as its confidence that a generated pair comes from the expert, and fed back to guide the policy. Raising the learning rate increased the reward, meaning that the pairs generated by the policy were judged closer to the attacker's. Like the losses, this reward measures how well the attacker is imitated and says nothing about the victim's ability to detect the attack.

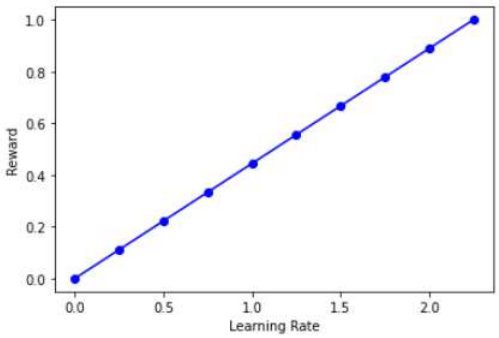


Fig. 7 Reward as a function of the learning rate in GAIL; a higher reward means that the generated pairs are judged closer to the attacker's demonstrations

4.3 Discussion

Both algorithms shed light on attacker behaviour, but in different ways. Behavior Cloning accurately reproduces the attacker's actions in most states and is simple to train, yet it is vulnerable to distribution shift and stumbles on the branching decisions at S_4, a known weakness of supervised imitation. GAIL learns the policy by running it in the environment rather than from the demonstrations alone, which in principle limits distribution shift; in our runs, however, the discriminator kept the upper hand over 50 epochs, so the imitation obtained with GAIL still needs to be improved. Neither model is a fraud detector: the GAIL discriminator compares the learned policy

with the attacker and never sees legitimate behaviour. What both models provide is an estimate of the attacker's strategy π_E. Combined with Eq. (9), this estimate can be used to study how a more cautious victim, for instance one with a higher $\rho(- \mid S_2, \cdot)$ after an awareness campaign, lowers the attacker's chance of reaching $S_\top$. These complementary views suggest that imitation learning is a viable, and arguably more natural, alternative to reinforcement learning for characterising mobile money phishing.

5 Conclusion and Future Work

We studied mobile money phishing through the lens of imitation learning, formalising the attacker–victim interaction as a four-state process driven by the attacker's actions and the victim's reactions, and applying Behavior Cloning and GAIL to attacker demonstrations. Behavior Cloning reproduced attacker actions well but was limited by distribution shift and by the complex decisions at the exit state. In GAIL, the discriminator separated the generated pairs from the attacker's demonstrations increasingly well, which shows that the imitation had not converged within 50 epochs; since this discriminator compares the policy with the attacker, it does not model a victim detecting phishing. These results point to imitation learning as a promising tool for understanding, and ultimately countering, mobile money phishing. The main limitations are the modest size of our demonstration dataset and GAIL's known issues of sample inefficiency, generator–discriminator balance, and reward definition. Future work will address these by collecting a larger expert dataset, by estimating the victim's strategy ρ from real complaint data so as to quantify the success probability of an attack, by training detectors on both legitimate and fraudulent interactions, and by exploring inverse reinforcement learning, which infers the reward function from observed behaviour rather than requiring it to be hand-designed.

Acknowledgements. The authors thank AIMS Rwanda and the Cybersecurity with Computational and Artificial Intelligence Research Group of the University of Ngaoundéré for their support.

Declarations

- **Funding.** No funding was received for conducting this study.
- **Competing interests.** The authors declare no competing interests.
- **Ethics approval and consent to participate.** Not applicable.
- **Consent for publication.** Not applicable.
- **Data availability.** The simulated datasets used in this study are available from the corresponding authors on reasonable request.
- **Code availability.** The code used in this study is available from the corresponding authors on reasonable request.
- **Author contributions.** A.N.N.: conceptualization, methodology, extension and improvement of the initial study, writing – review & editing. C.N.: initial investigation, software, formal analysis, writing – original draft. F.T.: conceptualization,

supervision, validation, writing – review & editing. D.M.B.: writing – review & editing, preparation of the submission. M.S.T.: formal analysis (formalisation of the attacker–victim interaction model), writing – review & editing. All authors read and approved the final manuscript.

References

[1] Muhire, A.I., Abayo, S., Assefa, T., Muvunyi, B.M., Liouane, M.K., Kasaazi, G.W., Mahoro, P.M., Iradukunda, P.: Mitigating mobile money services frauds in Rwanda. Journal of Cybersecurity **7**(1), 008 (2021)

[2] Quartz Africa: 70% of the world's $1 trillion mobile money market is in Africa. https://qz.com/africa/2161960/. Accessed: 2023-04-17 (2021)

[3] Kamande, M.W., Kamanzi, A., Kituyi, A., Qureshi, F.: Exploring the use of mobile money services among tea SACCOs in Rwanda: Challenges and opportunities. African Journal of Science and Technology (2020)

[4] International Finance Corporation: Rwanda market scoping report. Technical report, International Finance Corporation (2021). Accessed: 2023-04-14

[5] DW News: Rwanda to start new cashless era after COVID-19. DW News (2020)

[6] Venkatesha, S., Reddy, K.R., Chandavarkar, B.R.: Social engineering attacks during the COVID-19 pandemic. SN Computer Science **2**, 1–9 (2021)

[7] Mambina, I.S., Ndibwile, J.D., Michael, K.F.: Classifying Swahili smishing attacks for mobile money users: A machine-learning approach. IEEE Access **10**, 83061–83074 (2022)

[8] Chetioui, K., Bah, B., Alami, A.O., Bahnasse, A.: Overview of social engineering attacks on social networks. Procedia Computer Science **198**, 656–661 (2022)

[9] Tchakounte, F., Nyassi, V.S., Danga, D.E.H., Udagepola, K.P., Atemkeng, M.: A game theoretical model for anticipating email spear-phishing strategies. EAI Endorsed Transactions on Scalable Information Systems **8**(30), 5 (2021)

[10] The New Times: RIB warns of increased threat of mobile money fraud. The New Times (2020)

[11] Nzeket Njoya, A., Tchaya Ngongag, V.L., Tchakounte, F., Atemkeng, M., Fachkha, C.: Characterizing mobile money phishing using reinforcement learning. IEEE Access **11** (2023) https://doi.org/10.1109/ACCESS.2023.3317692

[12] Al-Otaibi, A.F., Alsuwat, E.S.: A study on social engineering attacks: Phishing attack. International Journal of Recent Advances in Multidisciplinary Research **7**(11), 6374–6380 (2020)

[13] Aldawood, H., Skinner, G.: A taxonomy for social engineering attacks via personal devices. International Journal of Computer Applications (2019)

[14] Salahdine, F., Kaabouch, N.: Social engineering attacks: A survey. Future Internet **11**, 89 (2019)

[15] Tandale, K.D., Pawar, S.N.: Different types of phishing attacks and detection techniques: A review. In: 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), pp. 295–299. IEEE (2020)

[16] Syafitri, W., Shukur, Z., Mokhtar, U.A., Sulaiman, R., Ibrahim, M.A.: Social engineering attacks prevention: A systematic literature review. IEEE Access **10**, 39325–39343 (2022) https://doi.org/10.1109/ACCESS.2022.3162594

[17] Hamaidi, K.: A predictive model for phishing detection based on convolutional neural networks. Master's thesis, University of Badji Mokhtar (2021)

[18] Bhavsar, V., Kadlak, A., Sharma, S.: Study on phishing attacks. International Journal of Computer Applications **182**, 27–29 (2018)

[19] Ho, J., Gupta, J., Ermon, S.: Model-free imitation learning with policy optimization. In: International Conference on Machine Learning, pp. 2760–2769. PMLR (2016)

[20] Zai, A., Brown, B.: Deep Reinforcement Learning in Action. Manning Publications (2020)

[21] Hussein, A., Gaber, M.M., Elyan, E., Jayne, C.: Imitation learning: A survey of learning methods. ACM Computing Surveys (CSUR) **50**(2), 1–35 (2017)

[22] Attia, A., Dayan, S.: Global overview of imitation learning. arXiv preprint arXiv:1801.06503 (2018)

[23] Torabi, F., Warnell, G., Stone, P.: Behavioral cloning from observation. arXiv preprint arXiv:1805.01954 (2018)

[24] Samak, T.V., Samak, C.V., Kandhasamy, S.: Robust behavioral cloning for autonomous vehicles using end-to-end imitation learning. arXiv preprint arXiv:2010.04767 (2020)

[25] Ho, J., Ermon, S.: Generative adversarial imitation learning. Advances in Neural Information Processing Systems **29** (2016)

[26] Gonfalonieri, A.: Generative adversarial imitation learning: Advantages & limits. Towards Data Science (2021)

[27] Akomea-Frimpong, I., Andoh, C., Akomea-Frimpong, A., Dwomoh-Okudzeto, Y.: Control of fraud on mobile money services in Ghana: an exploratory study. Journal of Money Laundering Control **22**(2), 300–317 (2019)

[28] Mohammed, A.: Mobile money SMS fraud detection. International Journal of Computer Applications **181**(43), 39–43 (2019)

[29] Botchey, F.E., Qin, Z., Hughes-Lartey, K.: Mobile money fraud prediction—a cross-case analysis on the efficiency of support vector machines, gradient boosted decision trees, and naïve Bayes algorithms. Information **11**(8), 383 (2020)

[30] Lokanan, M.: Predicting mobile money transaction fraud using machine learning algorithms. Qeios (2022)

[31] El Ayeb, S., Hemery, B., Jeanne, F., Cherrier, E.: Community detection for mobile money fraud detection. In: 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 1–6. IEEE (2020)

[32] Chebii, P.J.: Securing mobile money payment and transfer applications against smishing and vishing social engineering attacks. Master's research project, University of Nairobi (2021)

[33] Khan, S.: Human in the loop reinforcement learning framework for adaptive phishing detection. Discover Computing **28**, 311 (2025) https://doi.org/10.1007/s10791-025-09849-y

[34] Sharafat, A., Aymaz, S., Köse, C.: A hybrid CNN-BiGRU-attention architecture for enhanced spam email classification. Discover Computing **29**, 437 (2026) https://doi.org/10.1007/s10791-026-10102-3