MUSTER: Subverting User Selection in MU-MIMO Networks

Tao Hou[†]*, Shengping Bi[‡]*, Tao Wang[‡], Zhuo Lu[†], Yao Liu[†], Satyajayant Misra[‡], and Yalin Sagduyu[§]
[†]University of South Florida, Tampa, FL, USA, {taohou@, zhuolu@, yliu@cse.}usf.edu
[‡]New Mexico State University, Las Cruces, NM, USA, {sbi@, taow@, misra@cs.}nmsu.edu
[§]Intelligent Automation Inc, Rockville, MD, USA, ysagduyu@i-a-i.com
* Co-First Authors

***Abstract*—WiFi 5/6 relies on a key feature, Multi-User Multiple-In-Multiple-Out (MU-MIMO), to offer high-volume network throughput and spectrum efficiency. MU-MIMO uses a user selection algorithm, based on each user's channel state information (CSI), to schedule transmission opportunities for a group of users to maximize the service quality and efficiency. In this paper, we discover that such algorithm creates a subtle attack surface for attackers to subvert user selection in MU-MIMO, causing severe disruptions in today's wireless networks. We develop a system, named *MU-MIMO user selection strategy inference and subversion (MUSTER)*, to systematically study the attack strategies and further to seek efficient mitigation. MUSTER is designed to include two major modules: (i) strategy inference, which leverages a new neural group-learning strategy named MC-grouping via combining Recurrent Neural Network (RNN) and Monte Carlo Tree Search (MCTS) to reverse-engineer a user selection algorithm, and (ii) user selection subversion, which proactively fabricates CSI to manipulate user selection results for disruption. Experimental evaluation shows that MUSTER achieves a high accuracy rate around 98.6% in user selection prediction and effectively launches the attacks to disrupt the network performance. Finally, we create a Reciprocal Consistency Checking technique to defend against the proposed attacks to secure MU-MIMO user selection.**

I. Introduction

Multi-User Multiple-In-Multiple-Out (MU-MIMO), as an essential part of WiFi 5/6, has been widely supported in commercial wireless devices (e.g., WiFi routers/access points, mobile devices) to substantially improve the spectrum efficiency and increase the data throughput in wireless networks. MIMO indicates multiple propagation paths between the transmitter and receiver. To benefit from such spatial multiplexing, MU-MIMO allows the transmitter to send concurrent traffic streams to multiple receivers at the same time.

In practice, a transmitter is usually equipped with a limited number of antennas (e.g., up to 8 in WiFi 6 [1]), but the number of users can go up to tens or hundreds in MU-MIMO networks. Considering that the transmitter can only serve a small group of users at each data transmission session, how to select users to serve plays a crucial role to implement MU-MIMO networks. As concurrent data streams traveling through different propagation paths experience independent channel fading and may interfere each other, traditional user scheduling (e.g., round-robin user selection) without consideration of channel state information (CSI) may not obtain a user group with the least inter-user interference and is not suitable for MU-MIMO networks [2]. Recently, multiple CSI-based schemes have been proposed to achieve the optimal user selections and maximize the system throughput [3]–[8].

To achieve a more accurate and reliable CSI, implicit channel feedback is dropped in favor of explicit feedback in MU-MIMO networks [9]. Specifically, downlink CSI is estimated at each user and is then sent to the transmitter as the feedback for user selection. CSI is time-sensitive and may only remain consistent for a short time period. A timely channel feedback is critical to achieve fast and responsive communications. Research indicates that a 200ms feedback delay will result in a 50% degradation of achievable throughput in MU-MIMO networks [10]. Accordingly, CSI is required to be reported in plaintext as soon as possible [11] (e.g., plaintext feedback in WiFi 5/6).

Nevertheless, we discover that this convenient CSI feedback mechanism actually creates a subtle attack surface for attackers to subvert the user selection in MU-MIMO networks. Specifically, since the CSI feedback is self-reported and is transmitted in plaintext, an attacker is able to collect and analyze users' feedbacks, and further to delicately fabricate a forged channel feedback to manipulate the user selection results. In this work, we aim to investigate the potential attacks against CSI-based user selection algorithms, reveal the impacts of such attacks, and derive corresponding countermeasures to improve the security of MU-MIMO networks.

To this end, we present a strategy, named *MU-MIMO user selection strategy inference and subversion (MUSTER)*, that allows us to systematically study the potential risks of the user selection procedure and further to seek efficient mitigation. Specifically, we find that such a vulnerability may lead to three major categories of attacks that can essentially disrupt the CSI-based user selection from both user fairness and system throughput, which are the key objectives of implementing MU-MIMO networks.

- **Targeted Denial of Service (TDoS):** The attacker aims to starve particular users, such that the victims can never or barely get access to the transmitter. Such attacks can specify any victims to amplify its adverse impact, such as disconnecting important users who provide essential services, disrupting users requesting time-sensitive accesses, or starving local-network competitors.
- **Cooperative Privilege Escalation (CPE):** The attacker aims to escalate the privilege of particular users (e.g., a conspirator), increasing their possibility of being selected and obtaining exclusive service. In this way, the attacker

and the conspirator can cooperatively gain unfair access to the transmitter and abuse network operations.

- **Network Throughput Degradation (NTD):** One of the key objectives of user selection algorithms is to select a user group that achieves the maximum network throughput. By fabricating a forged CSI feedback, the attacker can subvert user selection results and substantially degrade the target MU-MIMO network throughput.

We adopt MUSTER to examine existing CSI-based user selection algorithms by exploiting these attacks, and reveal the potential risks. In the development of MUSTER, we meet two essential technical challenges that must be addressed to facilitate the attacker's purpose.

(1) *Comprehending the user selection strategy in black-box settings:* As not being specified in existing standards (i.e., WiFi 5/6), current practices of user selection algorithms for MU-MIMO transmission are vendor-implementation specific [12]. These implementations are confidential or proprietary in commercial products. Without the knowledge of how a transmitter selects users in such black-box settings, it is difficult for the attacker to craft the proposed attacks by scattershot approaches. To address this challenge, MUSTER introduces a design, named *User Selection Strategy Inference*, to proactively comprehend the user selection strategy of a target MU-MIMO network. In particular, we design a novel neural group-learning strategy, *MC-grouping*, that integrates Recurrent Neural Network (RNN) and Monte Carlo Tree Search (MCTS). It jointly considers users within possible groups yet substantially reduces the search space. With MC-grouping, the attacker can accurately predict user selection results and further subtly launch attacks.

(2) *Fabricating CSI feedbacks to subvert the user selection:* It is challenging to launch the proposed attacks merely with the predicted user selection results. We don't rely on adversarial perturbations [13], as such attacks without taking care of MIMO spatial compatibility can destroy the orthogonality property of precoded wireless communication and make messages not decodable at receivers. Instead, we propose an algorithm named *Spatial Compatibility Quantization* to learn the inter-user correlation among the predicted selection group in MUSTER. By learning the inter-user correlation among the predicted user group, we propose detail strategies on how to delicately fabricate the CSI feedback to launch each type of the potential attacks.

In addition to the attack strategy design, analysis, and evaluation, we develop an effective approach, named *Reciprocal Consistency Checking*, to protect the user selection from been undermined.

We implement MUSTER as a practical system and conduct experiments on real-world MU-MIMO networks with different user selection algorithms and settings. The experimental results show that the proposed strategy inference can accurately learn the user selection algorithm and achieve an accuracy rate up to 98.6% of user selection predictions. We also investigate the proposed attacks on top of user selection predictions. Results shows that TDoS can achieve up to 97.48% success rate, CPE can achieve up to 94.86% success rate, and NTD can substantially leads to 34.7%$\sim$ 54.3% network throughput degradation. The experiment results indicate that MUSTER can effectively launch desired attacks. We also study the proposed countermeasure to mitigate the discovered CSI-feedback related vulnerability. Our experiment shows that the proposed Reciprocal Consistency Checking can achieve a detection rate of 99.32%, essentially eliminating the potential attacks.

II. Preliminaries and Models

In this section, we briefly introduce the basic knowledge of MU-MIMO technique and the general user selection strategy.

A. Network Models

We consider a downlink MU-MIMO network with one base station transmitting to K users. The base station is equipped with N antennas and each user has one antenna. We denote the CSI vector between user i and the transmitter as $\mathbf{h_i}$=$[h_{i1}, h_{i2}, ..., h_{iN}]$, where h_{ij} indicates the CSI between user i and the transmitter's j^{th} antenna. Assume $K > N$ and a group of N users will be selected for the downlink MU-MIMO transmission.

In the MU-MIMO network, precoding needs to be applied at the base station to ensure that signals received at each user can be decoded independently. We focus on the widely used linear precoding strategy, zero-forcing beamforming (ZF-BF), which is adopted by WiFi 5/6. ZF-BF can eliminate the multi-user interference, thus allowing data to be decoded individually at each user.

Assume N users have been selected for ZF-BF. Denote the channel matrix of selected users as $\mathbf{H} = [\mathbf{h}_1^T, \mathbf{h}_2^T, ..., \mathbf{h}_N^T]^T$ (operator $\cdot^T$ denotes the matrix transpose). Denote the transmit physical-layer symbols over N antennas as $\mathbf{m} = [m_1, m_2, ..., m_N]^T$ with unit power. Then, the received signal vector of ZF-BF can be expressed as $\mathbf{y} = \mathbf{HWPm} + \mathbf{n}$, where $\mathbf{W}$ is the precoding matrix, $\mathbf{P}$ is the power loading diagonal matrix, and $\mathbf{n}$ is the channel noise vector. Let $\mathbf{w}_i = [w_{i1}, w_{i2}, \cdots, w_{iN}]^T$ denote the beamforming weight vector for user i. The precoding matrix is thus denoted as $[\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N]$.

The received signal for selected user i after precoding can be written as $y_i = \mathbf{h}_i\mathbf{w}_i\sqrt{p_i}m_i + \sum_{j\neq i}\mathbf{h}_i\mathbf{w}_j\sqrt{p_j}m_j + n_i$, where the first term $\mathbf{h}_i\mathbf{w}_i\sqrt{p_i}m_i$ is the desired signal, the second term $\sum_{j\neq i}\mathbf{h}_i\mathbf{w}_j\sqrt{p_j}m_j$ is the interference from concurrent signals, and the last term n_i is the noise. The corresponding Signal-to-Interference-plus-Noise Ratio (SINR) at user i can be represented as $\text{SINR}_i = \frac{|\mathbf{h}_i\mathbf{w}_i|^2 p_i}{\sum_{j\neq i}|\mathbf{h}_i\mathbf{w}_j|^2 p_j + \delta_i^2}$, where δ_i^2 is the channel noise power. In ZF-BF, given all CSI feedbacks, the base station knows $\mathbf{H}$ and constructs the precoding matrix $\mathbf{W}$ as $\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$, such that the interference $\mathbf{h}_i\mathbf{w}_j\sqrt{p_j}m_j$ is close to zero in practice and only $\mathbf{h}_i\mathbf{w}_i\sqrt{p_i}m_i$ remains at the receiver to be decoded.

Figure 1 presents an example of ZF-BF in a 2×2 MU-MIMO system: the precoded message $\mathbf{w_1}\sqrt{p_1}m_1$ is orthogonal to the channel $\mathbf{h_2}$, resulting in no interference to the receiver Rx_2 (i.e., $\mathbf{h_2w_1}\sqrt{p_1}m_1 = 0$). Meanwhile,

m_1 is decodable at receiver Rx_1 (i.e. $\mathbf{h_1}\mathbf{w_1}\sqrt{p_1}m_1 = |\mathbf{h_1}||\mathbf{w_1}\sqrt{p_1}|m_1\cos\theta$, where θ is the angle between $\mathbf{h_1}$ and $\mathbf{w_1}\sqrt{p_1}m_1$ in the vector space). Intuitively, when channel $\mathbf{h_1}$ is orthogonal to $\mathbf{h_2}$, $\mathbf{h_1}\mathbf{p_1}m_1 = |\mathbf{h_1}||\mathbf{p_1}|m_1$, which can yield the largest SINR. By contrast, when $\mathbf{h_1}$ is aligned to $\mathbf{h_2}$, $\mathbf{h_1}\mathbf{p_1}m_1 = 0$, and no message can be decoded.

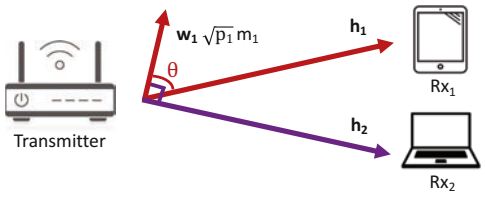

Fig. 1. An example of ZF-BF in a 2×2 MU-MIMO system.

Therefore, channel spatial compatibility, which reflects how well users' channels are orthogonal to each other [14], is a key feature to find a user group that yields the maximum system throughput. Specifically, users whose channels are spatially compatible to each other should be selected together to maximize the data throughput of the system.

B. User Selection Strategy

User selection for MU-MIMO is generally formulated as a sum rate (i.e., network throughput) maximization problem with varying constraints (e.g., bandwidth limitation, fairness) [15]. Let $\mathcal{K} = \{1, 2, \cdots, K\}$ denote the set of all users. The primary goal of the base station is to find a user set $\mathcal{C}$ ($\mathcal{C} \subset \mathcal{K}$) to maximize the total sum rate of the system (i.e., $\max_{\mathcal{C}\subset\mathcal{K}, 1\le|\mathcal{C}|\le N} \max_{\mathbf{W},\mathbf{P}} \sum_{i\in\mathcal{C}} \log(1+\text{SINR}_i)$, where $\mathbf{W}$ and $\mathbf{P}$ denote the precoding weights and power allocations.). The optimal strategy is to apply the brute-force exhaustive search over all possible user sets and find the one that yields the maximum data throughput. However, the cost of exhaustive search is exponentially expensive. Multiple greedy search based or heuristic schemes have been proposed to achieve the near-optimum yet efficient user selection [3]–[7].

III. Overview of MUSTER

In this section, we describe the discovered CSI feedback vulnerability and the overview of MUSTER.

Since explicit CSI feedback is used in today's MU-MIMO networks, a malicious user is able to report a fabricated CSI to the base station. This opens a door for exploiting the user selection algorithm at the base station to serve its malicious purposes. We classify such attacks into three categories: 1) *Targeted Denial of Service*; 2) *Cooperative Privilege Escalation*; 3) *Network Throughput Degradation*.

It is nontrivial to achieve each of the above attacks, because the attacker has no knowledge of the user selection algorithm and settings used in the network. In a MU-MIMO network, the user selection algorithm is vendor-dependent and proprietary (e.g., most commercial WiFi drivers are closed-source). Though the underlying principle of all user selection algorithms is to resolve the aggregated sum rate maximization, their implementations may be different to balance the performance and cost. It is necessary for the attacker to know the specific user selection algorithm in advance.

Intuitively, the attacker can adopt approaches of adversarial machine learning [16] to learn the inner structure of the target algorithm and create adversarial perturbations to subvert the user selection results. Due to the broadcast nature and open-text protocol, all CSI feedbacks are broadcast to the wireless channel. An attacker can decode them and treat them as the inputs for the black-box user selection algorithm. At the same time, the attacker can also observe which users have been selected in the open channel and treat the results as the outputs. Accordingly, the attacker can learn the input-output relationship to establish a substitute model for the user selection algorithm and further launch attacks.

Nevertheless, a closer examination shows that approaches initiated by existing wireless adversarial machine learning [17]–[19] cannot be readily adapted to comprehend and attack the user selection algorithm in MU-MIMO networks:

(1) The decision rules of existing approaches are usually binary, e.g., spoofing or no spoofing, jamming or no jamming. However, the possible outputs of user selection algorithms grow exponentially when the number of users increase. It is time-consuming or even computationally infeasible to enumerate all possible groups and portray their boundaries for a target user selection algorithm. Meanwhile, traditional machine learning usually treats each instance as an independent entity, they may not be able to capture the inter-user correlation.

(2) Creating adversarial perturbations is a computationally intensive tasks especially for a model with complicated decision boundaries, which is not feasible for the MU-MIMO network requiring prompt feedback. Further, adversarial perturbations without taking care of MIMO spatial compatibility can destroy the orthogonality property of precoded wireless communication and make messages not decodable at receivers, leading to an anomaly that can be easily detected.

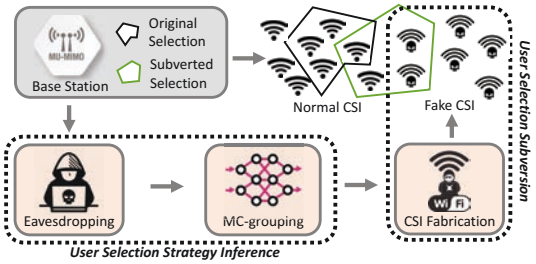

Fig. 2. The architecture of MUSTER.

These become the underlying motivation for us to derive a new way to subvert the user selection. Specifically, we propose MUSTER to address the above two issues. As shown in Figure 2, MUSTER consists of two major modules: (i) user selection strategy inference that allows the attacker to build a deep learning model with accurate prediction of user selection; (ii) user selection subversion that enables the attacker to effectively fabricate CSI feedbacks to achieve its malicious objectives. We introduce the technical details of modules (i) and (ii) in Sections IV and V, respectively.

IV. User Selection Strategy Inference

In this section, we design the strategy inference in MUSTER to reverse-engineer a target user selection algorithm.

A. Problem Statement

We denote the target user selection algorithm as M. The inputs of M are all K users' CSI feedbacks denoted as $\mathcal{H} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_K\}$, and other factors (e.g., bandwidth limitation, power allocation, utilization frequency) denoted as $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_K\}$, where $\mathbf{f}_k$ is the vector of other factors for user k. The corresponding output is the selected user group $\mathbf{G} = (a_1, a_2, ...a_N)$, where $N < K$ and a_n is the index of user n. The goal of strategy inference is to develop a deep-learning architecture, adapted to the input-output relation of the target black-box model M, such that producing the same selection results for the same user inputs.

As discussed in Section III, existing models used in wireless adversarial machine learning cannot be readily adapted to learn the MU-MIMO user selection algorithms. Intuitively, we can develop a group-based learning architecture to learn the user selection algorithms. Rather than treating a user as an independent entity, we can construct a model that examines all possible groups. However, such a trivial approach still incurs a considerable amount of computational overhead as it works in a brute-force manner. For a MU-MIMO network with N transmit antennas and K users, there are $\binom{K}{N}$ possible groups in total. Assume the computational complexity for examining each group is $O(N)$, the overall computational overhead would be $O(N\frac{K!}{(K-N)!})$, which significantly impedes the prediction efficiency especially when K is large.

To solve this, we aim to develop a new neural group-learning strategy that can jointly consider users within possible groups yet substantially reduce the search space.

B. A New Neural Group-learning Strategy

We observe that existing MU-MIMO user selection algorithms are usually statistical and heuristic, where users are selected step by step (e.g., only one user is selected at each step) [3]–[7]. It well maps to the property of Markov Decision Process (MDP). Inspired by that, we formulate the user selection procedure as an MDP. As shown in Figure 3, it starts with an empty group. At each level of the tree, it makes a decision and selects one user into the group until it reaches a leaf node. Each trajectory from the root to the leaf is a possible selection group.

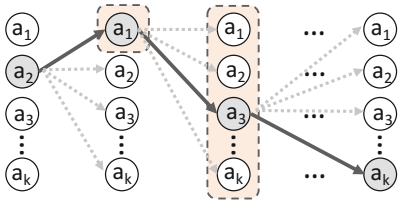

Fig. 3. User selection procedure as an MDP.

Nevertheless, it still face two challenges to apply MDP in user selection strategy inference: 1) In MDP, the decision on user to be selected next is only dependent on current step but is conditionally independent of all previous steps. However, the entire history of selected users needs to be considered to understand how inter-user correlations are used in the target user selection algorithm. 2) The MDP simplifies the user group prediction using decision policy but does not necessarily reduce the searching space of the model training. It is time-consuming to traverse all possible trajectories and train the model in a brute-force manner.

To this end, we develop a new neural learning strategy on the top of MDP, named *MC-grouping*. It combines a deep Recurrent Neural Network (RNN) [20] and Monte Carlo Tree Search (MCTS) [21] to jointly model users selected in a group to learn the decision policy, yet substantially reducing the search space of training. In particular, MC-grouping consists of two components: 1) RNN encoder: it utilizes an RNN architecture to encode the whole selection procedure (i.e., possible selected group and corresponding user inputs) into vector representations, such that selected users will be examined altogether. 2) MCTS based model training: it incorporates MCTS with RNN to expand the tree search with users that can yield promising rewards only, thereby reducing the search space of the possible groups.

C. MC-Grouping Design

In what follows, we present detailed designs of RNN encoding and MCTS based model training.

1) MDP Modeling: Mathematically, an MDP can be defined by a tuple $(\mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R})$, where $\mathbf{S}$ is the set of states, $\mathbf{A}$ is the set of actions, $\mathbf{P}$ is the state transition probability, and $\mathbf{R}$ is the reward function. In particular, as user selection is the core part of our strategy, $\mathbf{A}$ is specifically defined as the set of user candidates that can be selected in each level. Let $s_i \in \mathbf{S}$ denote the selection state at level i. Note that the entire history of selected users are required to learn the inter-user correlations, we thus define s_i as

$$s_i = q_{i-1} \cup \{a_{k,i}, \mathbf{h}_{\mathbf{a_{k,i}}}, \mathbf{f}_{\mathbf{a_{k,i}}}, \mathbf{K}_{i+1}, \mathbf{E}_{i+1}\}, \quad (1)$$

where q_{i-1} is defined as the traversed history of previous selected users, $a_{k,i}$ is the user selected at level i, $\mathbf{h}_{\mathbf{a_{k,i}}}$ and $\mathbf{f}_{\mathbf{a_{k,i}}}$ are the inputs of CSI feedback and other factors of user $a_{k,i}$, $\mathbf{K}_{i+1}$ is the set of user candidates in the next level, and $\mathbf{E}_{i+1}$ is the corresponding input of candidates (i.e., $\mathbf{E}_{i+1} = \{\mathbf{H}_{i+1}, \mathbf{F}_{i+1}\}$). s_i includes all essential information to describe current state of the selection procedure, 1) all current selected users along with their CSI feedbacks and other factors, 2) user candidates in next level. The selection terminates when it reaches the leaf node and outputs users of the entire trajectory as the predicted user group. We define the reward as $+1$ when the output includes the same selected users as the training data and 0 otherwise.

We further define the decision policy $\pi_\theta(a_k|s_i)$ as the probability of selecting user a_k given the the state s_i and Q function $Q_\theta(a_k|s_i)$ as the long term reward of taking user a_k given state s_i, where θ is a set of model parameters. The objective of the MDP is to learn a decision policy $\pi_\theta(a_k|s_i)$

and long-term reward $Q_\theta(a_k|s_i)$ that maximize the terminal rewards, i.e., to correctly identify the trajectory that contains the same selected user group as the target algorithm.

2) RNN Encoding: We create an RNN encoder, $\mathbf{G}s_i = Gen_\theta(s_i)$ to map the selection procedure into vector representations. θ is a set of RNN parameters (i.e., $\theta = \{\theta_U, \theta_G, \theta_Q, \theta_S\}$) as shown in Figure 4. We further rewrite s_i in (1) as, $s_i = q_i \cup \{\mathbf{K}_{i+1}, \mathbf{E}_{i+1}\}$. Accordingly, s_i consists of two parts, 1) $\{\mathbf{K}_{i+1}, \mathbf{E}_{i+1}\}$, which indicates user candidates to be selected in the next level; 2) q_i, which indicates the selection history and includes all selected users. We encode both components respectively.

Encoding candidate: As shown in Figure 4, we apply two concatenated fully connected layers to encode the whole candidates. The first layer f_{θ_U} is to encode each candidate a'_k in $\mathbf{K}_{i+1}$. Specifically, the vector representation $\mathbf{H}_{a'_k,i+1}$ of user a'_k is described as $f_{\theta_U}(a'_k, \mathbf{h}_{\mathbf{a}'_\mathbf{k}}, \mathbf{f}_{\mathbf{a}'_\mathbf{k}})$. With the parameter set θ_U, we can learn how CSI feedback and other factors are weighted to characterize one candidate in the target algorithm. The second layer f_{θ_G} is to summarize all the candidates in s_i, denoted as $\mathbf{G}_{K,i+1} = f_{\theta_G}(\mathbf{U}_{a'_1,i+1}, ..., \mathbf{U}_{a'_K,i+1})$.

Encoding selection history: q_i can be defined as a recursive function, $q_i = q_{i-1} \cup \{a_i, \mathbf{h}_{\mathbf{a_i}}, \mathbf{f}_{\mathbf{a_i}}\}$. We then use RNN to encode the selection history into vector representation. As shown in Figure 4, q_{i+1} is encoded as $\mathbf{R}_{q_{i+1}} = f_{\theta_Q}(\mathbf{R}_{q_i}, \mathbf{H}_{a_i,i}, \mathbf{G}_{K,i+1})$, where θ_Q is the model parameter, and $\mathbf{H}_{a_i,i}$ is the vector representation of selected user a_i. Finally, we apply another fully connected layer f_{θ_S} to map $\mathbf{R}_{q_{i+1}}$ into the vector representation, $\mathbf{V}_{S,i+1} = f_{\theta_S}(\mathbf{R}_{q_{i+1}})$, where θ_S is the parameter of the layer.

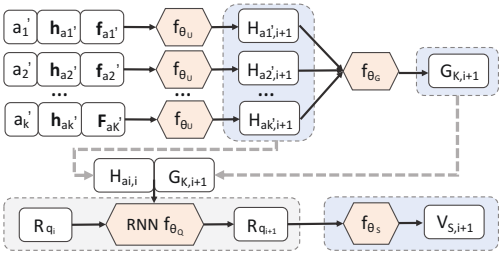

Fig. 4. RNN encoding.

The decision policy π_θ and Q-function Q_θ are jointly modeled by the inner product between $\mathbf{V}_{S,i+1}$ and $\mathbf{H}_{a'_k,i+1}$.

3) MCTS based Model Training: MCTS is a heuristic search algorithm introduced for computer Go (e.g., Google's AlphaGo) [22]. It can narrow down the search to the high probability selections, while still making close to optimal decisions at each step. We combine MCTS with RNN to train the parameter set θ (i.e., $\theta_U, \theta_G, \theta_Q, \theta_S$).

θ is updated in a policy iteration procedure. First, we run multiple MCTS simulations. Each simulation starts from the root state and iteratively selects users until it reaches the leaf node. In MC-grouping, we follow the upper confidence bound principle [23] to simulate the searching trajectory. After multiple simulations, we obtain an improved search policy that prefers users with larger accumulated rewards. Next, we learn from the the improved policy and update the parameter set θ to maximize the similarity between the improved policy and raw decision policy π_θ. As the improved search policy does not follow the original decision policy, θ is updated in an off-policy manner via Q-learning. We iteratively update θ until MC-Grouping can accurately predict the selected group via the policy π_θ.

V. User Selection Subversion

With MC-Grouping, MUSTER can accurately predict the user group who would be selected by the base station. However, how to efficiently launch attacks that can compromise the user selection results is still unclear. In this section, we aim to address this challenge.

As discussed in Section III, adversarial perturbations are not feasible to subvert the user selection in MU-MIMO networks. MUSTER develops new attack strategies by learning the spatial compatibility within the predicted user group. In what follows, we denote the predicted group as $\mathbf{S} = \{s_1, s_2, ..., s_N\}$, where s_i is the index of the selected user. The malicious user is denoted as s_a.

A. Targeted Denial of Service (TDoS)

In TDoS, the malicious user attempts to starve a particular user such that the victim can never or barely get service from the base station.

1) Attack Overview: Assume the targeted victim is user s_v ($s_v \in \mathbf{S}$). The malicious user aims to fabricate a forged CSI feedback, such that it will be selected and replace the victim in the selected user group. Intuitively, the malicious user may simply fabricate a forged channel feedback $\mathbf{h_a}$ as the proportional amplification of the victim's channel $\mathbf{h_v}$ (i.e., $\mathbf{h_a} = a\mathbf{h_v}$ and $|a| > 1$). In this way, $\mathbf{h_a}$ is aligned with the victim's channel $\mathbf{h_v}$ and has a larger channel gain.

Nevertheless, such a naive attack strategy may not always work. An increasing channel gain also indicates a stronger inter-user interference. The victim may still be selected because of the smaller interference. Further, different users should experience uncorrelated channels. It is impossible that channels at two users always align to each other [24]. The base station can easily detect such an attack if the attacker always feeds back an aligned channel with the victim's.

We design a practical attack strategy that can effectively starve a particular user. The strategy follows two principles: (1) The fabricated channel should have a larger channel gain but at the same time maintain a smaller inter-user interference; (2) The fabricated channel should impose the least impact on the user selection results (i.e., only the victim will be replaced in the selected user group). To this end, we first design an algorithm that can identify user's effective channel which is spatially compatible with others. Based on the victim's effective channel, the malicious user then fabricates a channel feedback that has a higher spatial compatibility within the select group and replace the victim.

2) Spatial Compatibility Quantization: The algorithm is designed to learn the inter-user correlation within the referred selected group and quantify users' effective channel. The

algorithm takes each user's CSI feedback in the predicted selected group $\mathbf{S}$ as input, quantifies their spatial compatibility and decomposes each channel as effective and interference parts. The detail design of *spatial compatibility quantization* is described in Algorithm 1. Note that the algorithm can be easily extended to the scenario when other factors (e.g., fairness scheduling) are considered.

Algorithm 1 Spatial compatibility quantization

Step 1 Initially, let $\mathbf{S'} = \mathbf{S}$, $\mathbf{S_0} = \varnothing$ and $i = 1$.

Step 2 For each user s_n in $\mathbf{S'}$, we first calculate the component $\mathbf{g_{(s_n)}}$ that is orthogonal with all users in $\mathbf{S_0}$,

$$\mathbf{g_{(s_n)}} = \mathbf{h_{(s_n)}} - \sum_{j=1}^{i-1} \frac{\mathbf{h_{(s_n)}} \mathbf{g_j^* g_j}}{||\mathbf{g_j}||^2}.$$

Note that when $i = 1$, $\mathbf{g_{(s_n)}} = \mathbf{h_{(s_n)}}$.

Step 3 Quantify the channel indexed by $\hat{s_n}$. Specifically,

$$\hat{s_n} = arg \max_{s_n \in \mathbf{S'}} ||\mathbf{g_{(s_n)}}||;\ \mathbf{g_i} = \mathbf{g_{(\hat{s_n})}};$$

$$\mathbf{e_i} = \sum_{\mathbf{j=1}}^{\mathbf{i-1}} \frac{\mathbf{h_{(\hat{s_n})}} \mathbf{g_j^* g_j}}{||\mathbf{g_j}||^2};\ \mathbf{S_0} = \mathbf{S_0} + \{\hat{s_n}\};$$

$$\mathbf{S'} = \mathbf{S'} - \{\hat{s_n}\};\ i = i + 1.$$

If $\mathbf{S'} \neq \varnothing$, then go to Step 2. Otherwise, algorithm stopes.

At shown, each user's channel $\mathbf{h_{(s_n)}}, s_n \in \mathbf{S}$ is decomposed as two parts, $\mathbf{g_{(s_n)}}$ and $\mathbf{e_{(s_n)}} = \sum_{\mathbf{j=1}}^{\mathbf{i-1}} \frac{\mathbf{h_{(s_n)}} \mathbf{g_j^* g_j}}{||\mathbf{g_j}||^2}$, where $\mathbf{g_{(s_n)}}$ is orthogonal with other users and is considered as the effective channel of the user, and $\mathbf{e_{(s_n)}}$ indicates the inter-user interference. The algorithm consumes vey limited computational resource as it only examine users within the selected group.

3) Attack Design: We apply spatial compatibility quantization algorithm over the referred select group $\mathbf{S}$ and decompose the target victim's channel $\mathbf{h_v}$ as $\mathbf{h_v} = \mathbf{g_v} + \sum_{i=1}^{v-1} \frac{\mathbf{h_v g_i^* g_i}}{||\mathbf{g_i}||^2}$, where $\mathbf{g_v}$ is the effective channel of the victim, and $\sum_{i=1}^{v-1} \frac{\mathbf{h_v g_i^* g_i}}{||\mathbf{g_i}||^2}$ is the component of inter-user interference. To effectively replace the victim in the selected user group, the attacker fabricates a channel with a larger effective channel gain but a smaller interference component. Specifically, the channel feedback can be fabricated as $\mathbf{h_a} = \alpha \mathbf{g_v} + \sum_{i=1}^{v-1} \omega_i \frac{\mathbf{h_v g_i^* g_i}}{||\mathbf{g_i}||^2}$, where α and ω_i are coefficients with $|\alpha| > 1$ and $|\omega_i| < 1$. Then, the effective channel of the fabricated feedback is now aligned with the victim's. They cannot be selected together because of the severe inter-user interference. As the fabricated feedback maintains a larger effective channel gain but a smaller interference component, the attacker will very likely replace the victim in the select group.

B. Cooperative Privilege Escalation (CPE)

In CPE, the malicious user attempts to manipulate the selection results to assist a particular user (i.e., a conspirator) to gain a higher possibility of being selected, achieving the exclusive access to the resources at the base station.

1) Attack Overview: Assume the conspirator is denoted as s_c. The malicious user aims to escalate its privilege, increasing the possibility of being selected. Intuitively, the conspirator can directly launch the TDoS to gain the exclusive service by fabricating a forged CSI feedback $\mathbf{h_c}'$. However, even if the conspirator successfully gets selected, it can hardly decode the message due to the inconsistence between the forged reported CSI and its genuine channel.

2) Attack Design: We develop a strategy that significantly increases the possibility of the conspirator accessing the service but does not require any modification on the conspirator's CSI feedback. Specifically, the malicious user is involved to help the conspirator. First, we apply the spatial compatibility quantization over the select group to find out effective channel of each selected user and get a ranked group $\mathbf{S_0}$. Second, following the reverse order in $\mathbf{S_0}$, we compare the conspirator with each selected user according to their orthogonal and interference components, and find the possible victim j who can be replaced by the conspirator. Third, the malicious user launches the TDoS attack to replace the $j-1^{th}$ selected user. Specifically, the feedback $\mathbf{h_a}$ fabricated by the malicious user is orthogonal to the conspirator but significantly interfered with the victim. In this way, when selecting next user, the conspirator will have an escalated chance to be selected as its effective channel gain now is larger than the victim's.

C. Network Throughput Degradation (NTD)

The objective of a user selection algorithm is to select a user group that can achieve the maximum network throughout gain. Nevertheless, we demonstrate that a malicious user can substantially degrade the network throughput.

1) Attack Overview: In NTD, the malicious user attempts to fabricate a CSI feedback to subvert the selection results, diminishing the effective network throughput of the selected group. Intuitively, the overall resources (e.g., bandwidth, number of users served, power) at the base station are fixed, the attacker can fabricate a CSI feedback to acquire as many resources as possible, such that the legitimate users can only obtain limited resources, yielding a lower effective network throughput. Here, we define the effective network throughput as the achievable sum rate of all selected legitimated users. The malicious user degrades the throughput from two perspectives: 1) decreasing the effective channel gain of selected users, 2) increasing the inter-user interference among selected users.

2) Attack Design: The attack is composed of two parts: 1) Explicit throughput degradation: The malicious user takes advantage of the TDoS attack to replace the user with the maximum effective channel gain in the predicted group, directly degrading the effective network throughput. 2) Implicit throughput degradation: The malicious user deliberately crafts a CSI feedback that interferes with users in the predicted group, such that these users are replaced by other tendentious users to degrade overall network throughput.

VI. Experimental Evaluation

We build a real-world 4×18 MU-MIMO system and implement multiple typical user selection algorithms to evaluate the

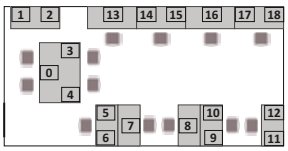

Fig. 5. Floor plan.

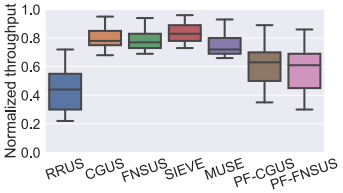

Fig. 6. Throughput distribution.

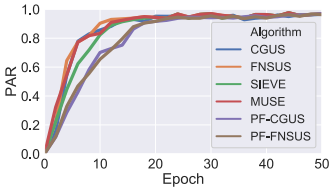

Fig. 7. PARs of different algorithms.

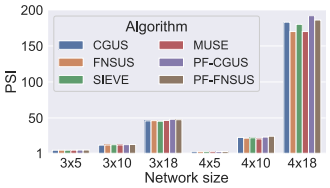

Fig. 8. PSIs for different networks.

proposed attack strategies.

A. Experiment Setup

The system is built on top of the universal software radio peripheral (USRP) following WiFi 6 standard [25]. USRP is a software defined radio device capable of implementing different MAC-layer and physical layer designs. The base station is built with four USRPs synchronized via an external clock OctoClock-G, which can distribute a high-accurate time scale and clock reference. The clients are standalone USRPs. Figure 5 exhibits the floor plan of the experiment. The base station is located at position 0 and the clients are located at position 1∼18. Without loss of generality, the one at position 6 is malicious.

B. User Selection Algorithms

Table I lists user selection algorithms implemented in the system. We compare the performance of these algorithms via the achievable network throughput. In particular, we repeat each algorithm for 1000 times and measure their network throughput (all results are normalized by the throughput of optimal user selection). Figure 6 box-plots the distribution of their throughput. As shown, SIEVE, CGUS, FNSUS and MUSE have the comparable performance and can achieve 72% $\sim$ 83% of the optimal throughput on average, while RRUS has the worst performance, resulting in more than 55% throughput degradation on average. When fairness scheduling is enforced, both CGUS and FNSUS suffer throughout loss in return for fairness. The results indicate that CSI based user selection algorithms indeed improve the network throughput in MU-MIMO networks.

TABLE I
USER SELECTION ALGORITHMS IN MU-MIMO NETWORK.

#	Algorithm	Description
1	OUS	Brute-force search.
2	RRUS	Each user is equally selected in a circular order.
3	CGUS	CGUS iteratively selects users as long as the aggregated network throughput improves.
4	FNSUS	FNSUS iteratively adds new users according to the defined orthogonality criterion.
5	SIEVE	SIEVE iteratively refines the candidate set via the branch-and-bound tree searching.
6	MUSE	MUSE identifies the inter-user correlation and select users with compressed CSI feedback.

C. Evaluation of Strategy Inference

We evaluate the proposed strategy inference from both aspects of the prediction accuracy and computational overhead.

1) Data Collection and Evaluation Metrics: Data collection is done via passive eavesdropping by the malicious user. For each user selection algorithm, we collect 11K pairs of user feedbacks and corresponding selection results, with 10K pairs for training and 1K pairs for testing.

In the experiment, we follow the strategy of the MC-grouping to build the deep learning model for each user selection algorithm, and then evaluate its prediction accuracy and computational overhead. Specifically, we define two metrics for our evaluation: 1) Prediction Accuracy Rate (PAR): PAR is to measure the effectiveness of the proposed strategy inference. It is defined as $PAR = \frac{Number\ of\ correct\ predictions}{Number\ of\ total\ predictions}$. 2) Prediction Speed Improvement (PSI): PSI is to measure the prediction efficiency of the proposed strategy inference by comparing it with a brute-force learning model that examines all possible groups. The metric is defined as $PSI = \frac{Time\ overhead\ of\ brute-force\ learning}{Time\ overhead\ of\ MC-grouping}$.

2) Prediction Accuracy: We evaluate the prediction accuracy of strategy inference for all CSI based user selection algorithms listed in Table I. PAR is measured after each epoch (i.e., one cycle through the full training dataset). The evaluation results are exhibited in Figure 7. As shown PARs for all algorithms can approach a high accuracy around 98.6% after a certain number of epochs. For example, PAR for CGUS can reach stability after 16 epochs. This observation indicates that strategy inference can effectively learn different user selection algorithms and get accurate user selection predictions.

3) Prediction Efficiency: In the evaluation, we use PSI to illustrate the performance improvement of the proposed strategy over the brute-force learning in various network sizes. As shown in Figure 8, the proposed MC-grouping strategy outperforms the brute-force learning in all different size of networks. When the network is small (i.e., 3x5, 4x5), the proposed strategy is 1.2 $\sim$ 2.5 times faster than the brute-force learning. As the computational overhead of the brute-force learning increases exponentially when the number of users increase, the proposed strategy can achieve orders of magnitude faster prediction when the network is large. e.g., the prediction of the proposed strategy is 178 times faster than brute-force learning for CGUS in the 4x18 MU-MIMO network. The results demonstrate that the proposed MC-grouping strategy indeed provides an efficient and salable user selection prediction especially for a large MU-MIMO network.

D. Evaluation of User Selection Subversion

We evaluate the effectiveness of the proposed attacks under different user selection algorithms and settings.

1) Target Denial of Service: For each algorithm, MUSTER first builds a deep learning model to accurately predicts the selected user group. Then we repeats each algorithm for 1000 times. Every time when the victim (user 2 or 15) is selected in predicted group, the malicious user becomes active and launch the TDoS to replace the victim in the group. Otherwise, the malicious user behaves as a passive eavesdropper.

We define following three metrics to evaluate the effectiveness of the attack: 1) NS_{orig}: It is defined as the number of times the victim originally been selected among 1000 executions. For a fair algorithm, each user is selected for 222 (i.e., $\frac{4\times1000}{18}$) times on average; 2) NS_{TDoS}: It is defined as the number of times the victim been selected among 1000 executions when the proposed attack is present. 3) RS_{succ}: The metric indicates the success rate of the attack of TDoS. It is defined as $RS_{succ} = \frac{NS_{orig}-NS_{TDoS}}{NS_{orig}}$.

Table II illustrates the performance of the TDoS under different scenarios. When TDoS is not present, all the algorithms can achieve a relatively fair results (i.e. NS_{orig} is raging from 199 to 241). When TDoS is launched, it can achieve a success rate up to 97.48%, essentially starving the victim. Meanwhile, depending on how fairness scheduling is configured, TDoS can still cut off more than 60% or 40% of opportunities of the victim been serviced in the network. It seems fairness scheduling can be applied to resolve the attack of TDoS. Nevertheless, fairness is achieved at the cost of network throughput. A heavily weighed fairness scheduling will considerably degrade the overall throughput of the MU-MIMO network. It's always a trade-off between the fairness and network throughput.

TABLE II
EFFECTIVENESS OF TDOS (θ IS THE COEFFICIENT OF PROPORTIONAL FAIRNESS, POS. INDICATES THE POSITION OF THE VICTIM.)

Alg.	θ	Pos.	NS_{orig}	NS_{sta}	RS_{succ}
CGUS	0	2	199	5	97.48%
CGUS	0	15	241	8	96.68%
FNSUS	0	2	237	7	97.04%
FNSUS	0	15	209	6	97.12%
SIEVE	0	2	222	7	96.84%
SIEVE	0	15	231	6	97.40%
MUSE	0	2	201	7	96.51%
MUSE	0	15	213	8	96.24%
PF-CGUS	1	2	215	83	61.39%
PF-CGUS	5	2	229	127	44.54%
PF-FNSUS	1	2	230	81	64.78%
PF-FNSUS	5	2	228	135	40.79%

2) Cooperative Privilege Escalation: In the evaluation, we repeat each algorithm for 1000 times. When the conspirator (user 4 or 16) is not selected in the predicted group, the malicious user actively launch the CPE attack to escalate the conspirator's possibility of being selected. Similarly, we define four metrics to evaluate the effectiveness of the attack: 1) N_{orig}: The metric is defined as the number of times the conspirator originally been selected among 1000 executions; 2) N_{pri}: The metric is defined as the number of times the conspirator been selected when launching the attack of CPE; 3) R_{succ}: The metric indicates the success rate of the attack. It is defined as $R_{succ} = \frac{N_{pri}-N_{orig}}{1000-N_{orig}}$; 4) PDR: The metric measures the average package delivery rate when the conspirator has been selected. For an effective CPE, it should not impact the message decoding at the conspirator.

Table III illustrates the performance of the CPE under different scenarios. As shown, the attack of CPE does not impact message decoding at the conspirator (i.e., PDR is always higher than 97.17%). When CPE is not present, all the algorithms can achieve a relatively fair results (i.e. N_{orig} is raging from 201 to 243). When CPE is launched, it can achieve more than 91.16% success rate, essentially escalating the privilege of the conspirator. Meanwhile, when fairness scheduling is applied, it can achieve a success rate up to 49.35% given different fairness coefficients. The possible reason is that when a user has been selected in a line, the corresponding user throughput weighted by the fairness coefficient will be dramatically reduced, leaving the user even harder to be selected. To alleviate the problem, the conspirator can change his/her identify (via IP spoofing or MAC spoofing) every time he/she gets served, such that the weighted throughput will be refreshed and reset to the default value.

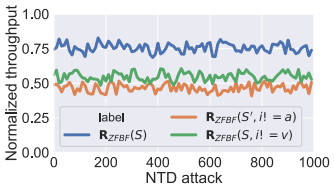

Fig. 9. The attack of NTD in CGUS.

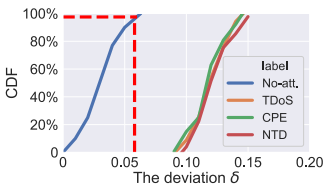

Fig. 10. Distribution of δ.

TABLE III
EFFECTIVENESS OF CPE.

Alg.	θ	Pos.	N_{orig}	N_{pri}	R_{succ}	PDR
CGUS	0	4	231	932	91.16%	98.34%
CGUS	0	16	201	957	94.62%	99.13%
FNSUS	0	4	241	961	94.86%	99.17%
FNSUS	0	16	232	953	93.88%	98.13%
SIEVE	0	4	225	942	92.52%	98.09%
SIEVE	0	16	229	949	93.38%	98.93%
MUSE	0	4	201	943	92.87%	99.01%
MUSE	0	16	243	954	93.92%	97.89%
PF-CGUS	1	4	226	579	45.61%	97.17%
PF-CGUS	5	4	215	321	13.50%	98.23%
PF-FNSUS	1	4	234	612	49.35%	97.20%
PF-FNSUS	5	4	213	330	14.87%	98.76%

3) Network Throughput Degradation: In the evaluation, we repeat each algorithm for 1000 times. The malicious user aims to downgrade the network performance by 1) explicitly replacing a user with the largest effective channel gain; 2) implicitly user group replacement. To this end, we evaluate the performance of NTD by comparing three types of network throughput, 1) $R_{\text{ZFBF}}(\mathbf{S})$, the achievable network throughput of the original predicted group $\mathbf{S}$; 2) $R_{\text{ZFBF}}(\mathbf{S}', i \neq a)$, the effective throughput of the manipulated group (i.e., the malicious user is excluded); 3) $R_{\text{ZFBF}}(\mathbf{S}, i \neq v)$, the throughput of original group excluding the user with largest channel gain.

Our experiments reveal that NTD can effectively degrade the MU-MIMO network throughput. Figure 9 give an example of NTD attack in CGUS. As shown, NTD can leads to a 34.7% $\sim$ 54.3% network throughput degradation, where about 25% $\sim$ 40% degradation comes from the explicitly user replacement while 5% $\sim$ 18% comes from implicitly group substitution.

VII. COUNTERMEASURES

Intuitively, we can encrypt the CSI feedback to keep it confidential from malicious users. However, adopting modern ciphers (e.g., AES) at the physical layer incurs more complexity and CSI feedback delay that can degrade the MU-MIMO

performance [26]. Alternatively, we propose a lightweight yet effective approach, named *Reciprocal Consistency Checking*, to protect the user selection in MU-MIMO networks.

A. Reciprocal Consistency Checking

This approach aims to detect a forged CSI feedback by exploiting channel reciprocity of downlink and uplink signals. Due to the channel reciprocity, transceivers of the same wireless link should observe the similar channel. Inspired by the property, we may detect a forged CSI feedback by checking the consistency between the uplink channel estimated by the base station and the downlink channel feedback from the user. However, this straightforward approach may not work due to imbalanced amplitude attenuations and phase rotations in channels introduced by hardware circuit modules at the base station and users. In particular, assume the CSI feedback of user i is denoted as $\mathbf{h_f} = [h_{f1}, h_{f2}, ..., h_{fN}]$ and the channel estimated at the base station is denoted as $\mathbf{h_e} = [h_{e1}, h_{e2}, ..., h_{eN}]$. Ideally h_{fn} should be equal to h_{en} (i.e., $\frac{h_{fn}}{h_{en}} = 1$), but they could be quite different because of the imbalanced hardware-oriented distortions.

Nevertheless, we observe that the hardware-oriented distortion is identical for channels estimated at the same device (i.e., $h_{f1}, ..., h_{fN}$ experience the same hardware-oriented distortions) [9]. Though $\frac{h_{fn}}{h_{en}} \neq 1$, the ratios between any pair of (h_{fn}, h_{en}) should be equal to each other, i.e., $\frac{h_{f1}}{h_{e1}} = \frac{h_{f2}}{h_{e2}} = ... = \frac{h_{fN}}{h_{eN}}$. Hereby, we develop a lightweight fake CSI detection scheme by checking the consistency among all the pair ratios of $\frac{h_{fn}}{h_{en}}$ for any $n \in \{1, 2, ...N\}$. Specifically, we use the variance δ as the metric to indicate the deviations among ratios of $\frac{h_{fn}}{h_{en}}$, $\delta = \frac{1}{N}\sum_{n=1}^{N}(\frac{h_{fn}}{h_{en}} - \overline{\left(\frac{h_f}{h_e}\right)})^2$, where $\overline{\left(\frac{h_f}{h_e}\right)}$ is the average of all channel ratios between the base station and users. When the channel feedback $\mathbf{h_f}$ is genuine, the deviation of $\frac{h_{fn}}{h_{en}}$ only comes from the channel noise and imperfect time synchronization. δ should be very small. Meanwhile, when $\mathbf{h_f}$ is deliberately manipulated to modify the user selection results, the ratios of $\frac{h_{fn}}{h_{en}}$ will not be consistent with each other, resulting in a larger δ.

B. Experimental Evaluation

We collect 1000 pairs of CSI feedbacks from multiple clients and the corresponding channel estimations at the base station for different scenarios (i.e., attack or non-attack). Let $\mathbf{h_{fi}}$ denote the i^{th} channel feedback and $\mathbf{h_{ei}}$ denote the corresponding local channel estimation. For each pair of $\mathbf{h_{fi}}$ and $\mathbf{h_{ei}}$, their ratio deviation δ is defined as, $\delta_i = \frac{1}{4}\sum_{n=1}^{4}|\frac{h_{fn}}{h_{en}} - \overline{\left(\frac{h_f}{h_e}\right)}|^2$. Figure 10 exhibits the distribution of channel ratio deviations. As shown, when CSI feedback is genuine, δ is very small, more than 90% of deviations are less than 0.05. Meanwhile, when an attack is present, more than 95% of deviations are larger than 0.1. Particularly, we can set the empirical threshold τ as 0.06 to determine a fake channel feedback, such that we can achieve a detection rate of 99.32% while only have a false positive rate of 0.05%.

VIII. Related Work

Recently, research has been initiated to understand the feasibility and impacts of attacks leveraging adversarial machine learning in different wireless scenarios, including wireless signal spoofing [27], [28], spectrum poisoning [17], [18], and smart jamming attacks [29], [30]. Most scenarios considered in these initial studies share a common characteristic: the output of a decision rule to be learned by the attacker is binary (e.g., spoofing or no spoofing [17], jamming or no jamming [19], spectrum available or unavailable [18]). The binary output makes it relatively simple to train a machine learning model. Unfortunately, these traditional models cannot be readily adapted to learn MU-MIMO user selection as its decision rules featuring the number of outputs grows exponentially when the number of users increases. It is expensive or even computationally infeasible to enumerate and learn decision boundaries for all possible groups by directly adopting a common machine learning based classifier. As a result, the user selection strategy inference in MUSTER integrates RNN and MCTS that can jointly consider users within possible groups yet substantially reduce the search space. Furthermore, in contrast to existing studies [31], MUSTER does not rely on adversarial perturbations to launch attacks because such attacks affects spatial compatibility among multiple users and lead to undecodable communication.

Our work is also related to research that exploits explicit plaintext CSI feedback. In [11], the authors present a sniffing attack that allows an attacker to eavesdrop concurrent data streams of victims by reporting a crafted CSI feedback. A formal mathematical analysis has been present in [9] to model the CSI-forgery based eavesdropping attacks. The work in [32] further refines the attack by optimizing the eavesdropping opportunity of attackers. Those attacks target on compromising the data confidentiality and integrity of MU-MIMO systems. By contrast, in this work, we develop MUSTER to exploit attacks on subverting user selection algorithms, compromising both user fairness and system throughput, which are two key objectives of implementing MU-MIMO networks.

IX. Conclusion

In this work, we propose a system, named MUSTER, to systematically study the potential risks of the user selection in MU-MIMO networks. The MUSTER system consists of two major modules: (i) strategy inference, which leverages a new neural group-learning strategy named MC-grouping via combining RNN and MCTS to reverse-engineer a user selection algorithm; (ii) user selection subversion, which proactively fabricates CSI to manipulate user selection results for disruption. We also develop a technique, Reciprocal Consistency Checking that can defend against aforementioned attacks to secure the user selection in MU-MIMO networks.

X. Acknowledgement

The authors at the University of South Florida were supported in part by NSF under grants CNS-2044516, ECCS-2029875 and CNS-1553304.

REFERENCES

[1] M Shahwaiz Afaqui, Eduard Garcia-Villegas, and Elena Lopez-Aguilera. Ieee 802.11 ax: Challenges and requirements for future high efficiency wifi. *IEEE Wireless Communications*, 24(3):130–137, 2016.

[2] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. Practical mu-mimo user selection on 802.11ac commodity networks. In *Proceedings of MobiCom '16*, page 122–134, New York, NY, USA, 2016.

[3] Shengchun Huang, Hao Yin, Jiangxing Wu, and Victor C. M. Leung. User selection for multiuser mimo downlink with zero-forcing beamforming. *IEEE Transactions on Vehicular Technology*, 62(7):3084–3097, 2013.

[4] Taesang Yoo and A. Goldsmith. On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming. *IEEE Journal on Selected Areas in Communications*, 24(3):528–541, 2006.

[5] G. Dimic and N.D. Sidiropoulos. On downlink beamforming with greedy user selection: performance analysis and a simple new algorithm. *IEEE Transactions on Signal Processing*, 53(10):3857–3868, 2005.

[6] Zukang Shen, Runhua Chen, J.G. Andrews, R.W. Heath, and B.L. Evans. Low complexity user selection algorithms for multiuser mimo systems with block diagonalization. *IEEE Transactions on Signal Processing*, 54(9):3658–3663, 2006.

[7] Jianqi Wang, David J. Love, and Michael D. Zoltowski. User selection with zero-forcing beamforming achieves the asymptotically optimal sum rate. *IEEE Transactions on Signal Processing*, 56(8):3713–3726, 2008.

[8] Xiufeng Xie and Xinyu Zhang. Scalable user selection for mu-mimo networks. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 808–816, 2014.

[9] Sulei Wang, Zhe Chen, Yuedong Xu, Qiben Yan, Chongbin Xu, and Xin Wang. On user selective eavesdropping attacks in mu-mimo: Csi forgery and countermeasure. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 1963–1971, 2019.

[10] Ehsan Aryafar, Narendra Anand, Theodoros Salonidis, and Edward W. Knightly. Design and experimental evaluation of multi-user beamforming in wireless lans. In *Proceedings of MobiCom '10*, page 197–208, New York, NY, USA, 2010. Association for Computing Machinery.

[11] Yu-Chih Tung, Sihui Han, Dongyao Chen, and Kang G. Shin. Vulnerability and protection of channel state information in multiuser mimo networks. In *Proceedings of CCS '14*, page 775–786, New York, NY, USA, 2014.

[12] Eng Hwee Ong, Jarkko Kneckt, Olli Alanen, Zheng Chang, Toni Huovinen, and Timo Nihtilä. Ieee 802.11 ac: Enhancements for very high throughput wlans. In *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 849–853. IEEE, 2011.

[13] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, 2016.

[14] Randolph Nelson and Leonard Kleinrock. Spatial tdma: A collision-free multihop channel access protocol. *IEEE Transactions on communications*, 33(9):934–944, 1985.

[15] Eduardo Castañeda, Adão Silva, Atílio Gameiro, and Marios Kountouris. An overview on resource allocation techniques for multi-user mimo systems. *IEEE Communications Surveys Tutorials*, 19(1):239–284, 2017.

[16] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 241–257, New York, NY, USA, 2019. Association for Computing Machinery.

[17] Tugba Erpek, Yalin E. Sagduyu, and Yi Shi. Deep learning for launching and mitigating wireless jamming attacks. *IEEE Transactions on Cognitive Communications and Networking*, 5(1):2–14, 2019.

[18] Zhengping Luo, Shangqing Zhao, Zhuo Lu, Jie Xu, and Yalin Sagduyu. When attackers meet ai: Learning-empowered attacks in cooperative spectrum sensing. *IEEE Transactions on Mobile Computing*, pages 1–1, 2020.

[19] Nguyen Van Huynh, Diep N. Nguyen, Dinh Thai Hoang, and Eryk Dutkiewicz. "jam me if you can:" defeating jammer with deep dueling neural network architecture and ambient backscattering augmented communications. *IEEE Journal on Selected Areas in Communications*, 37(11):2603–2620, 2019.

[20] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[21] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

[22] Michael C Fu. Alphago and monte carlo tree search: the simulation optimization perspective. In *Proceedings of IEEE WSC*, 2016.

[23] Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 225–240. Springer, 2013.

[24] Woongsup Lee, Juyeop Kim, and Sang-Won Choi. New d2d peer discovery scheme based on spatial correlation of wireless channel. *IEEE Transactions on Vehicular Technology*, 65(12):10120–10125, 2016.

[25] Universal software radio peripheral. https://en.wikipedia.org/wiki/Universal_Software_Radio_Peripheral.

[26] Serge Vaudenay. *A classical introduction to cryptography: Applications for communications security*. Springer Science & Business Media, 2006.

[27] Muhammad Zaid Hameed, András György, and Deniz Gündüz. The best defense is a good offense: Adversarial attacks to avoid modulation detection. *IEEE Transactions on Information Forensics and Security*, 16:1074–1087, 2021.

[28] Bryse Flowers, R. Michael Buehrer, and William C. Headley. Evaluating adversarial evasion attacks in the context of wireless communications. *IEEE Transactions on Information Forensics and Security*, 15:1102–1113, 2020.

[29] Yalin E. Sagduyu, Yi Shi, and Tugba Erpek. Iot network security from the perspective of adversarial deep learning. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2019.

[30] Yalin E. Sagduyu, Yi Shi, and Tugba Erpek. Adversarial deep learning for over-the-air spectrum poisoning attacks. *IEEE Transactions on Mobile Computing*, 20(2):306–319, 2021.

[31] Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*, 2018.

[32] Xiaoshan Wang, Yao Liu, Xiang Lu, Shichao Lv, Zhiqiang Shi, and Limin Sun. On eavesdropping attacks and countermeasures for mu-mimo systems. In *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, pages 40–45, 2017.