Secure Mobile Edge Computing in IoT via Collaborative Online Learning

Bingcong Li, Tianyi Chen, and Georgios B. Giannakis

***Abstract*—To accommodate heterogeneous tasks for the Internet of Things (IoT), the emerging mobile edge paradigm extends computing services from the cloud to the edge, but at the same time exposes new challenges on security. In this context, the present paper deals with online security-aware edge computing under jamming attacks. Leveraging online learning tools, novel approaches are developed to cope with adversarial worst-case attacks, and stochastic attacks with random attack strategies. Rather than relying on extra bandwidth and power resources to evade jamming attacks, the resultant algorithms select the most reliable server to offload computing tasks with minimal security concerns. It is analytically established that without any prior information on future jamming and server security risks over a time horizon T, the proposed schemes can achieve $\mathcal{O}\left(\sqrt{T}\right)$ regret. Information sharing among devices can accelerate the security-aware computing tasks, quantified by what is termed "value of cooperation." Effectiveness of the proposed schemes is tested on synthetic and real datasets.**

***Index Terms*—Cyber security, mobile edge computing, online learning, multi-armed bandit, jamming.**

I. Introduction

Internet of Things (IoT) impacts every aspect of daily life ranging from healthcare, video recognition in smart homes and smart cities, to monitoring the smart grids [11], [23]. Among these, various latency-sensitive applications such as autonomous driving and virtual reality raise new challenges to the current IoT paradigms. A critical one among these challenges is how to simultaneously meet the demands of huge volume and latency-sensitive data requests while acknowledging the limited computing power of IoT devices. As cloud computing alone cannot handle such IoT requirements, edge computing has emerged as a promising complement to subside the computational resources from the cloud to edge servers [31], which also facilitates real-time computing [8], [9], [11], [27], [35].

Although edge computing enables offloading computationally intensive tasks to the edge, security issues could prevent one from fully embracing its potential [14], [25], [30], [38]. As an example, although edge computing facilitates location based-services (e.g., social networking), the users' location information is also exposed to edge nodes, from which a "malicious" edge node can pry into users' private data [25]. In addition, data collected by a critical infrastructure such as the power grid need also to be veiled in order to prevent blackouts as well as malicious attacks. However, the high-complexity encryption techniques with complicated cipher-decipher processes are usually not computationally affordable by IoT devices (e.g., sensors in smart grids) [25]. One approach to coping with such privacy concerns is to allow devices to choose their trusted services on-demand with the so-called "transparent computing model" [28], while the trustworthiness of edge servers could be evaluated by the trust management services [26].

Besides privacy concerns, jamming and eavesdropping - the two main attacks at the physical layer [29], [42] - are still an issue for IoT systems. The present work mainly focuses on security-aware edge computing in the presence of jamming that can block the communication between IoT devices and edge servers; see [16] for eavesdropping issues in IoT.

Existing works dealing with jamming in IoT include those that detect anomalies [24], and those that mitigate jamming effects mainly using extra resources such as power and bandwidth [4], [12], [14], [30], [36], [39], [40]. Optimal power allocation schemes for jamming attacks were studied in [12]. Assuming a low-power jammer, game-theoretic anti-jamming strategies were reported in [14], [39]. However, for IoT devices with limited battery capacity, excess power consumption is not always affordable. On the other hand, spectrum allocation to evade jamming was considered in [30] for cognitive radio networks. Frequency hopping strategies without pre-shared secrets, a.k.a. uncoordinated frequency hopping, was adopted in [36], where a multi-armed bandit (MAB) scheme was introduced to allocate frequency bands only considering one transmitter-receiver pair; and in [4] for large-scale and more sophisticated cognitive radio networks. Recent efforts have been devoted to improving energy efficiency via channel hopping based anti-jamming schemes, e.g., [40], where an MAB channel selection scheme was proposed to maximize energy efficiency. However, spectrum expansion-based schemes have limited applicability for the spectrum-scarce IoT setups.

Jamming attacks can also be studied from a stochastic game point of view. A minimax Q-learning approach was studied in [34], but entails bandwidth expansion. When facing both jamming and eavesdropping, a game-theoretic approach was investigated in [13] see also [17], [41]. In addition, application-specific game-based approaches may not offer a good fit for the heterogeneous IoT devices, where the communication protocols vary across devices.

In this paper, we develop novel approaches to security-aware edge computing based on a non-stochastic MAB formulation

Work in this paper was supported by NSF 1509040, 1508993, 1711471, and 1901134.

B. Li, and G. B. Giannakis are with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA. Emails: {lixx5599, georgios}@umn.edu

T. Chen was with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA. He is now with Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, New York, 12180, USA. Email: chent18@rpi.edu.

[1], [2], [19], [20], where accessibility of an edge server is time-varying. In this context, each IoT device progressively learns the risk associated with edge servers, and adaptively chooses the most secure edge server to offload computing tasks among all available servers per slot. In contrast to [32], [33], [37], where the risks are assumed to follow some unknown distributions, the present does not make any assumption on how the adversarial attacker degrades the security. The time-varying servers can be dealt with using a MAB framework with sleeping arms [19], [20]. However, in the edge computing scenario with a bunch of devices, IoT devices are further envisioned sharing information to cooperatively secure edge computing. To account for the mobility and connectivity of IoT devices, communication links among devices will be modeled as time-varying wireless connections with security information shared among connected devices. The performance gain brought by cooperation in securing edge computing will be also rigorously established.

Our main contributions can be summarized as follows.

c1) Leveraging MAB tools, we develop two algorithms to deal with edge computing in the presence of adversarial and stochastic jamming attacks.

c2) We analytically establish that an $\mathcal{O}\big(\sqrt{T}\big)$ regret can be achieved by both proposed algorithms over a time horizon T, and benefit from device cooperation with markedly lower risks - a quantifiable performance gain of cooperation that we call the *value of cooperation*.

Notations. Bold lowercase letters denote column vectors; $\mathbb{E}$ denotes the expectation; $\mathbb{1}$ denotes the indicator function; and $(\cdot)^\top$ stands for vector transposition.

II. Modeling and Problem Statement

This section introduces the models and the formulation of the security-aware task in the presence of jamming attacks.

A. Modeling preliminaries

Consider an IoT scenario with a set of $\mathcal{K} := \{1, 2, \ldots, K\}$ edge servers to handle computational requests from a set of devices $\mathcal{J} := \{1, 2, \ldots, J\}$. Per slot t, the task of device j is described by the pair (c_t^j, s_t^j), where c_t^j captures the resources (e.g., CPU cycles) needed to complete the task, and s_t^j is the size of the computational task (including data input and the associated processing code) [6], [9].

Security risk. When an edge server is attacked, it can behave unfaithfully or intentionally sabotage the computation tasks. To cope with such a compromise on privacy, an IoT device needs to select the most reliable server for offloading computing tasks. To quantify an edge server's reliability, a commonly accepted metric is the security risk, which can be assessed using e.g., the number of attacks within a slot duration [26]. Per slot t, the security risk $r_t^j(k)$ of a server k can be observed by a device j only after this server completes the device's task. To break down $r_t^j(k)$, let $\gamma_{c,t}(k)$ denote the *unit* risk of computing at server k, and $\gamma_{s,t}(k)$ the *unit* risk for privacy information leakage at server k, which reflect the intensity of the attacks aiming to degrade the accuracy of computational results and private data, respectively. Means of inferring $\gamma_{c,t}(k)$ and $\gamma_{s,t}(k)$ can be found in [26] and the references therein. Since both $\gamma_{c,t}(k)$ and $\gamma_{s,t}(k)$ could be *adversarial* depending on the nature of attackers, we do not make any stochastic assumption on them. Furthermore, let $\rho^j \in [0,1]$ denote a device-specific weight on security; e.g., a larger ρ^j can be used for safety-sensitive tasks involved in autonomous vehicles and healthcare, while a smaller ρ^j can be adopted by sensors in smart homes and smart grids where privacy is the first priority. At the end of slot t, the risk for device j to choose a compromised server k is modeled by [5]

$$r_t^j(k) = \rho^j c_t^j \gamma_{c,t}(k) + (1-\rho^j) s_t^j \gamma_{s,t}(k) \tag{1}$$

where $r_t^j(k)$ captures the task-specific risk of receiving an inaccurate computational result and the risk of data leakage.

Jamming attacks. Malicious attackers can also sabotage the IoT devices by blocking their access to edge servers [29], [42]. To account for such jamming effects, we collect the accessible servers for device j in a set $\mathcal{K}_t^j \subseteq \mathcal{K}$, from which device j will select the most reliable server.

Jammers can be modeled as one of two types.

Adversarial jammers: these strategically attack edge servers, possibly taking into account the devices' past offloading decisions [14], and $\mathcal{K}_t^j$ can be *any* subset of $\mathcal{K}$; and,

Stochastic jammers: these attack IoT device-server links with a fixed probability; hence, each server has a fixed probability of being included in $\mathcal{K}_t^j$, meaning $\mathcal{K}_t^j$ is random.

For device j to know the available server set $\mathcal{K}_t^j$ it can securely offload its computational task, edge servers must broadcast a signal to make their accessibility known. Depending also on the jammer's attacking strategy, for two different devices j and j', it is possible that $\mathcal{K}_t^j \neq \mathcal{K}_t^{j'}$. For simplicity, we assume that the accessibility of servers remains unchanged over the duration of a slot. Otherwise, the server has to return the computational result in the following non-jammed slot, which can be afforded only in delay-tolerant settings [22].

Device cooperation. Information sharing among IoT devices can assist security-aware computing at the edge [10]. After device j observes $\gamma_{c,t}(k)$ and $\gamma_{s,t}(k)$ *at the end* of slot t, it can communicate this information to its one-hop neighbors. This set of one-hop neighboring devices can vary from slot to slot due to e.g., mobility. We suppose that information sharing is directional, meaning it is possible for j' to receive information from j, but not vice versa. During the information sharing stage, device j obtains the security risks of servers in a subset $\mathcal{S}_t^j \subseteq \mathcal{K}$. Note that $\mathcal{S}_t^j$ may include servers not in $\mathcal{K}_t^j$ since other devices may share the risk information of servers not belonging to $\mathcal{K}_t^j$. Let $a_t^j \in \mathcal{K}_t^j$ denote the index of the server assigned to carry out the computational task of device j in slot t. Other devices can select the same server chosen also by device j.

To summarize, per slot t device j performs three steps

s1) receives the information of the accessible $\mathcal{K}_t^j$;

s2) selects one edge server $a_t^j \in \mathcal{K}_t^j$ to offload; and

s3) observes $r_t^j(a_t^j)$, as well as shares (obtains) risk information with (from) its one-hop neighbors to form $\mathcal{S}_t^j$.

For simplicity, each server is presumed to have enough capacity to handle the computational load. This can be readily relaxed in practice by taking the following steps.

s1) A heavily loaded server introducing large delays can act as if it is jammed to avoid being chosen by IoT devices;

s2) The main challenge at the servers with moderate delay is that their risks are observed by IoT devices with a certain delay. MAB with delayed feedback can effectively address this issue; see e.g., [3], [18], [22].

The goal is for IoT devices to choose *online* (at the beginning of each slot t) the edge server minimizing the accumulated risk; that is,

$$\min_{\{a_t^j \in \mathcal{K}_t^j, \forall t, \forall j\}} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{j=1}^{J} r_t^j(k) \mathbb{1}(a_t^j = k) \tag{2}$$

Problem (2) is an integer program that cannot be solved efficiently, simply because each IoT device j must select a secure server a_t^j at the beginning of each slot t, before the corresponding risks $\{r_t^j(k)\}_{k=1}^K$ become available at the end of slot t. We will relax (2) but its relaxation will end up admitting the same optimum solution as that of (2).

In addition to solving (2) in the presence of jammers, we further wish to quantify the benefit of cooperation among devices. While the present formulation considers a setting without explicit cooperation constraints, it can be readily extended to incorporate long-term constraints; see e.g., [7].

B. Linear programming reformulation

Since $r_t^j(a_t^j)$ is revealed *after* the server a_t^j is chosen, it is reasonable for each device to select servers according to the past server risks, while allowing for flexibility in this choice. In par with this guideline, suppose that device j selects each of the K servers according to a $K \times 1$ probability mass function (pmf) vector having as kth entry the probability of selecting the server k; that is, $a_t^j \sim \mathbf{p}_t^j \in \mathbb{R}^K$. Rather than optimizing over K-valued variables $\{a_t^j\}$, problem (2) can be relaxed to minimize the *expected risk* over the pmf vector entries $\{p_t^j(k) := \Pr\{a_t^j = k\}\}_{k=1}^K$; that is,

$$\min_{\{\mathbf{p}_t^j \in \Delta(\mathcal{K}_t^j), \forall t, j\}} \sum_{t=1}^{T} \sum_{j=1}^{J} \left(\mathbf{r}_t^j\right)^\top \mathbf{p}_t^j \tag{3}$$

where the $\mathcal{K}_t^j$-related "probability simplex" is

$$\Delta(\mathcal{K}_t^j) := \left\{ \mathbf{p} \in \mathbb{R}_+^K \,\middle|\, \sum_{k \in \mathcal{K}_t^j} p(k) = 1;\, p(k)=0, k \notin \mathcal{K}_t^j \right\} \tag{4}$$

with $p(k)$ denoting the k-th entry of $\mathbf{p}$. The rationale behind (3) is that a randomized server selection scheme may have better worst-case performance in expectation than deterministic schemes [15, Theorem 1.1]. In addition, a deterministic server selection incurs higher risk since adversaries can potentially decipher the selection strategy in use and act strategically.

Problem (3) is a linear program separable per device j. Hence, it can be readily solved had we known the sequence $\{\mathbf{r}_t^j\}_{t=1}^T$ as well as the accessible server sets $\{\mathcal{K}_t^j\}_{t=1}^T$. To infer the unavailable risks and decide on the secure server, we will view the per-device accessible edge servers as arms in an MAB setting. However, the plain-vanilla MAB with a fixed set of arms is not applicable because the available servers here can change from slot to slot. As a result, the selected server according to $\mathbf{p}_t^j$ may not be accessible. In addition, the time-varying availability of servers also challenges the tradeoff between exploration (which promotes servers that have not been selected frequently) and exploitation (which favors the most secure server observed so far). This well known exploration-exploitation tradeoff shows up also in the 'workhorse' MAB solver [1]. However, the solver of (3) must further account for the need to adaptively choose the edge server according to a time-varying feasibility set $\mathbf{p}_t^j \in \Delta(\mathcal{K}_t^j)$.

III. Edge Computing under Adversarial Jamming

In this section, we introduce schemes for secure server selection when the adversary affects the available set of servers $\mathcal{K}_t^j$ per device j and slot t.

A. Reducing complexity using risk-ordered lists of servers

For the online learning problem (2) or (3), optimizing over $\{a_t^j\}$ or $\{p_t^j(k)\}$, can be viewed as learning a policy f that maps the set $\mathcal{K}_t^j$ to the optimal server $a_t^j = f(\mathcal{K}_t^j)$ with minimum risk, while obeying the server availability constraint $f(\mathcal{K}_t^j) \in \mathcal{K}_t^j$. When necessary, the sought function f can be device specific (denoted by f^j), but for convenience, we henceforth confine ourselves to a single f. Consider an instantiation of $\mathcal{K}_t^j$ denoted generically by the set $\tilde{\mathcal{K}}$ of non-jammed servers, with cardinality $|\tilde{\mathcal{K}}|$. Since we have K servers in total, and each one of them can be jammed or not, there are 2^K possible non-jammed server sets, namely $\tilde{\mathcal{K}}_1, \ldots, \tilde{\mathcal{K}}_{2^K}$. Thus, the number of possible policies is the product of the corresponding cardinalities, that is, $\prod_{i=1}^{2^K} |\tilde{\mathcal{K}}_i|$.

To alleviate the high complexity of finding the optimal f among candidate policies, we will build on the *risk-ordered list of servers* approach in [20]. Each such list is a permutation π_n of server indexes $\{1, \ldots, K\}$ listed according to a possible ordering of their risks with the lowest-risk server index listed first. If e.g., $K = 3$, one permutation can be $\pi_n = \{2, 3, 1\}$ signifying that the second server has the lowest risk, the third one has higher risk than the second, and the first server has the highest risk in this particular list. Clearly, a risk-ordered list specifies a policy outcome because with a given non-jammed server set $\tilde{\mathcal{K}}$, the optimal server index $k_n \in \tilde{\mathcal{K}}$, having the lowest risk in the list is the one appearing earliest in the list π_n. If in our example with $K = 3$ and $\pi_n = \{2, 3, 1\}$, the non-jammed servers set is $\tilde{\mathcal{K}} = \{1, 3\}$, the optimal server (or equivalently optimal policy for this list) is $k_n = f_n(\tilde{\mathcal{K}}) = 3$. With $\bar{\mathcal{K}}$ collecting all indices of permutations in $\mathcal{K}$, the cardinality of $\bar{\mathcal{K}}$ which is also the number of all policies $\bar{K} := |\bar{\mathcal{K}}| = K!$. Clearly, $K!$ is smaller than $\prod_{i=1}^{2^K} |\tilde{\mathcal{K}}_i|$. As the policy space formed by the risk-ordered list of policies is much smaller than the original space, the complexity can be significantly reduced. Next, we will show that the risk-ordered lists of policies also contain the optimal policy. For a given $\tilde{\mathcal{K}}$, consider the $\bar{K} \times K$ matrix $\mathbf{\Gamma}(\tilde{\mathcal{K}})$ with (n, k)-th entry

$$\left[\mathbf{\Gamma}(\tilde{\mathcal{K}})\right]_{n,k} = \mathbb{1}(f_n(\tilde{\mathcal{K}}) = k). \tag{5}$$

Accordingly, for adversarial jammers (3) can be rewritten using expanded $\bar{K} \times 1$ pmf vectors $\{\bar{\mathbf{p}}_t^j\}$ as

$$\min_{\bar{\mathbf{p}}_t^j \in \Delta^{\bar{K}}} \sum_{t=1}^{T} \sum_{j=1}^{J} \left(\bar{\mathbf{r}}_t^j\right)^\top \bar{\mathbf{p}}_t^j \tag{6}$$

where $\bar{\mathbf{r}}_t^j$ is a $\bar{K} \times 1$ vector given by $\bar{\mathbf{r}}_t^j = \boldsymbol{\Gamma}(\mathcal{K}_t^j)\mathbf{r}_t^j$; and the $\bar{K}$-dimensional probability simplex is $\Delta^{\bar{K}} := \big\{\bar{\mathbf{p}} \in \mathbb{R}_+^{\bar{K}} \,\big|\, \sum_{k\in\bar{\mathcal{K}}} \bar{p}(k) = 1\big\}$. The search in (6) is for a $\bar{K} \times 1$ vector $\bar{\mathbf{p}}_t^j$ that takes into account all risk-ordered lists in $\bar{\mathcal{K}}$. The next lemma establishes that (3) is equivalent to (6).

Lemma 1. *Given $\mathcal{K}_t^j$ and $\mathbf{p}_t^j \in \Delta(\mathcal{K}_t^j)$, there exists at least one $\bar{\mathbf{p}}_t^j \in \Delta^{\bar{K}}$, such that $\big(\bar{\mathbf{p}}_t^j\big)^\top \bar{\mathbf{r}}_t^j = \big(\mathbf{p}_t^j\big)^\top \mathbf{r}_t^j$.*

Proof. See Appendix A. □

Lemma 1 implies that if we search over the risk-ordered lists of servers (policies), the optimal solution coincides with that of the original problem. This motivates our reduced-complexity solver of (6) that we develop next.

B. Existing methods in adversarial jamming

The algorithm we introduce here to solve (6) builds on the exploration, exploitation, and exponential (EXP3) iteration developed in [1] for *non-cooperative* settings, where the set $\mathcal{K}_t^j$ is fixed across devices and slots. To tailor EXP3 to our setup with a time-varying arm set $\mathcal{K}_t^j$ per device j and slot t, the extended $\bar{K}$-dimensional formulation can be used, where each arm of the MAB (and thus EXP3) is a risk-ordered list. Indeed, a given extended pmf vector $\bar{\mathbf{p}}_t^j$ implies a corresponding risk-ordered list of servers (and thus induced policy) with list index variable $a_t^j \sim \bar{\mathbf{p}}_t^j$, and yields the non-jammed server with lowest risk as $a_t^j = f_{a_t^j}(\mathcal{K}_t^j)$. After the edge computing task is completed at slot t, device j observes only the risk of the list indexed by a_t^j, that is $\bar{r}_t^j(a_t^j) = r_t\big(f_{a_t^j}(\mathcal{K}_t^j)\big)$. To complement this partial observation of the risk, EXP3 relies on an importance sampling type of risk estimates given by

$$\hat{\bar{r}}_t^j(n) = \frac{\bar{r}_t^j(n)\mathbb{1}\big(a_t^j = n\big)}{\bar{p}_t^j(n)}, \ \forall n \in \bar{\mathcal{K}} \tag{7}$$

where $\bar{r}_t^j(n) = r_t^j\big(f_n(\mathcal{K}_t^j)\big)$. The denominator in (7) ensures unbiasedness, while the indicator function in the numerator implies that only one of $\bar{K}$ entries of the estimated risk vector is nonzero. We rely on (7) to find the next pmf $\bar{\mathbf{p}}_{t+1}^j$. But first, we will go after its unnormalized counterpart given by

$$\bar{\mathbf{w}}_{t+1}^j = \arg\min_{\bar{\mathbf{w}}} \eta(\hat{\bar{\mathbf{r}}}_t^j)^\top(\bar{\mathbf{w}} - \bar{\mathbf{w}}_t^j) + \mathcal{D}_{\mathrm{KL}}(\bar{\mathbf{w}}||\bar{\mathbf{w}}_t^j) \tag{8}$$

where the constant learning rate η controls the cost versus regularization provided by the KL-divergence $\mathcal{D}_{\mathrm{KL}}(\bar{\mathbf{w}}||\bar{\mathbf{w}}_t^j) := \sum_{n=1}^{\bar{K}} \bar{w}(n) \ln\big(\bar{w}(n)/\bar{w}_t^j(n)\big)$. Without the KL regularizer the cost approximates that in (6), but the resultant solution turns out to have inferior performance relative to the regularized iterate in (8) [1].

Having found $\bar{\mathbf{w}}_{t+1}^j$ in (8), the pmf iterate is obtained as

$$\bar{p}_{t+1}^j(n) = \frac{\bar{w}_{t+1}^j(n)}{\sum_{m\in\bar{\mathcal{K}}} \bar{w}_{t+1}^j(m)}\ , \qquad n \in \bar{\mathcal{K}}. \tag{9}$$

Differentiating per entry of $\bar{\mathbf{w}}$ in (8), and equating the result to zero yields readily a closed-form multiplicative update

$$\begin{aligned} \bar{w}_{t+1}^j(n) &= \bar{w}_t^j(n)\exp\Big(-\eta\hat{\bar{r}}_t^j(n)\Big)e^{-1} \qquad (10)\\ &= \exp\Big(-\eta\sum_{\tau=1}^{t}\hat{\bar{r}}_\tau^j(n)\Big)e^{-t}\ , \quad n \in \bar{\mathcal{K}}, t \geq 1 \end{aligned}$$

where for the second equality we used $\bar{w}_1^j(n) = 1, \forall n$. The intuition behind (10) is that the multiplicative iteration accumulates risk in the exponential, and thus exploits past experience. As a result, a smaller $\sum_{\tau=1}^t \hat{\bar{r}}_\tau^j(n)$ leads to a larger $\bar{p}_{t+1}^j(n)$. Besides, EXP3 implicitly controls the exploration-exploitation tradeoff, by letting $\bar{p}_{t+1}^j(a_t^j) \leq \bar{p}_t^j(a_t^j)$ and $\bar{p}_{t+1}^j(n) \geq \bar{p}_t^j(n), \forall n \neq a_t^j$; that is, exploring another server list $n \neq a_t^j$ in the next time slot, is encouraged by EXP3. Despite its simple implementation and performance guarantees [1], the efficiency of updating the expanded pmf vector can be significantly improved using [20]. However, cooperation that has not been exploited in [1], [20], will be leveraged in the ensuing subsection by our Security-Aware edge serVer sElection (SAVE-A) algorithm to better mitigate adversarial jamming effects.

C. SAVE-A for edge computing under adversarial jamming

Suppose that per slot t, device j first selects a risk-ordered list index α_t^j (policy $f_{\alpha_t^j}$) according to $\bar{\mathbf{p}}_t^j$, leading to a selected server index $a_t^j = f_{\alpha_t^j}\big(\mathcal{K}_t^j\big)$. Once the computing tasks are completed by the end of slot t, device j observes not only the risk of server a_t^j, but also the risk of servers in $\mathcal{S}_t^j$, thanks to cooperation. This means it is possible for device j to observe the risk of servers in $\tilde{\mathcal{K}}_t^j := \mathcal{K}_t^j \cup \mathcal{S}_t^j$. When evaluating the performance of different policies f_n per risk-ordered list π_n, set $\tilde{\mathcal{K}}_t^j$ will replace $\mathcal{K}_t^j$, because $\tilde{\mathcal{K}}_t^j$ is more informative than $\mathcal{K}_t^j$; that is, $\mathcal{K}_t^j \subseteq \tilde{\mathcal{K}}_t^j$.

Consider now that risk-ordered lists (and thus policies) belong to one of the following two complementary sets: i) $f_n(\tilde{\mathcal{K}}_t^j) \in \tilde{\mathcal{K}}_t^j \setminus \mathcal{S}_t^j$; and ii) $f_n\big(\tilde{\mathcal{K}}_t^j\big) \in \mathcal{S}_t^j$. For policies in i), the risk of the policy is revealed to device j if it is chosen by device j; while for policies in ii), the risk of the policy is revealed to device j regardless of which server device j selects, because server risks can be shared across devices. As a result, the risk estimators of policies in i) and ii) are constructed differently. For policies $f_n(\tilde{\mathcal{K}}_t^j) \in \tilde{\mathcal{K}}_t^j \setminus \mathcal{S}_t^j$, an interesting observation is that given $\tilde{\mathcal{K}}_t^j$, it is possible to have $f_n(\tilde{\mathcal{K}}_t^j) = f_m(\tilde{\mathcal{K}}_t^j)$. Hence, once we know the risk $\bar{r}_t^j(n)$, we also deduce $\bar{r}_t^j(m) = \bar{r}_t^j(n)$. This in turn suggests for $f_n\big(\tilde{\mathcal{K}}_t^j\big) \in \tilde{\mathcal{K}}_t^j \setminus \mathcal{S}_t^j$, the risk estimator

$$\hat{\bar{r}}_t^j(n) = \frac{\bar{r}_t^j(n)\mathbb{1}\big(f_n(\tilde{\mathcal{K}}_t^j) = a_t^j\big)}{\mu_t^j + \sum_{m=1}^{\bar{K}} \bar{p}_t^j(m)\mathbb{1}\big(f_m(\tilde{\mathcal{K}}_t^j) = f_n(\tilde{\mathcal{K}}_t^j)\big)}, \quad \forall n : f_n\big(\tilde{\mathcal{K}}_t^j\big) \in \tilde{\mathcal{K}}_t^j \setminus \mathcal{S}_t^j \tag{11a}$$

where μ_t^j in (11a) introduces bias, but it stabilizes the risk estimator when the second term in denominator is small. For policies with $f_n\big(\tilde{\mathcal{K}}_t^j\big) \in \mathcal{S}_t^j$, the counterpart of (11a) is

$$\hat{\bar{r}}_t^j(n) = \frac{\bar{r}_t^j(n)}{\mu_t^j + 1}, \quad \forall n : f_n\big(\tilde{\mathcal{K}}_t^j\big) \in \mathcal{S}_t^j. \tag{11b}$$

Since sharing risk information can be viewed as one kind of exploring other servers, μ_t^j is added to confine the exploration. To appreciate this, recall that the exploration in EXP3 is achieved by enforcing $\bar{p}_{t+1}^j(a_t^j) \leq \bar{p}_t^j(a_t^j)$, and a larger $\hat{\bar{r}}_t^j(a_t^j)$ leads to a larger $\bar{p}_t^j(a_t^j) - \bar{p}_{t+1}^j(a_t^j)$. Intuitively, by adding μ_t^j in the denominator, we manually reduce the value of $\hat{\bar{r}}_t^j(a_t^j)$ to

Algorithm 1 SAVE-A for IoT device j

```
1:  Initialize: weight $\mathbf{w}_1^j = \mathbf{1}/\bar{K}$, exploration factor $\mu_t^j$, and learning rate $\eta_t^j$.
2:  for $t = 1, 2, \ldots, T$ do
3:      Available server set $\mathcal{K}_t^j$ is revealed.
4:      Choose $f_{\alpha_t^j}(\cdot) \sim \bar{\mathbf{p}}_t^j$, and select $a_t^j = f_{\alpha_t^j}(\mathcal{K}_t^j)$,
5:      Receive $\gamma_{c,t}(a_t^j)$ and $\gamma_{s,t}(a_t^j)$.
6:      Broadcast $\gamma_{c,t}(a_t^j)$ and $\gamma_{s,t}(a_t^j)$ to devices in $\{i \mid j \in \mathcal{S}_t^i\}$.
7:      Compute security risk for $\{a_t^j\} \cup \mathcal{S}_t^j$ via (1).
8:      Estimate $\bar{\mathbf{r}}_t^j$ via (11).
9:      Update $\bar{\mathbf{w}}_{t+1}^j$ via (12) and compute $\bar{\mathbf{p}}_{t+1}^j$ via (13).
10: end for
```

lower $\bar{p}_t^j(a_t^j) - \bar{p}_{t+1}^j(a_t^j)$, thus confining the exploration. This intuition is further validated in Corollary 1 by choosing μ_t^j relatively large when $\{|\mathcal{S}_t^j|\}$ is large.

Using (11a) or (11b) to form $\hat{\bar{\mathbf{r}}}_t$, device j maintains an unnormalized weight $\bar{\mathbf{w}}_{t+1}^j \in \mathbb{R}^{\bar{K}}$ to evaluate the estimated historical cumulative security risks of policies $\{f_n\}$, that is

$$\bar{w}_{t+1}^j(n) = \exp\Big(-\eta_{t+1}^j \sum_{\tau=1}^{t} \hat{\bar{r}}_\tau^j(n)\Big), \ \forall n \in \bar{\mathcal{K}} \tag{12}$$

where $\sum_{\tau=1}^{t} \hat{\bar{r}}_\tau^j(n)$ is the estimated cumulative risk of policy f_n for device j. The difference between (12) and (10) is the device-specific time-varying stepsize η_{t+1}^j adopted by SAVE-A. The specific choice of η_{t+1}^j will be provided in Corollary 1. Intuitively, enhanced cooperation yields more reliable risk estimates in $\hat{\bar{\mathbf{r}}}_t^j$, which can afford a larger η_{t+1}^j.

Finally, device j obtains the pmf $\bar{\mathbf{p}}_{t+1}^j$ as in (9); that is

$$\bar{p}_{t+1}^j(n) = \frac{\bar{w}_{t+1}^j(n)}{\sum_{m \in \bar{\mathcal{K}}} \bar{w}_{t+1}^j(m)}. \tag{13}$$

The proposed SAVE-A is summarized in Algorithm 1.

D. Regret analysis for SAVE-A

An online algorithm is desirable for standard MAB settings when its regret is sublinear with respect to the time horizon T, written as $\mathrm{Reg}_T = o(T)$, where the Reg_T is defined as the accumulated risk of the per-slot optimal online solution over T slots minus the minimum risk of a fixed server in *hindsight* [2], [15]. A sublinear regret implies $\lim_{T\to\infty} \mathrm{Reg}_T/T = 0$ so that the algorithm is asymptotically not worse than choosing the best fixed server. However, the best fixed server used in [1], [2], [15] may not be always accessible in the presence of jamming. This necessitates resorting to the best fixed policy as a benchmark in our adversarial setup.

Among all $\{f_n\}$, the fixed policy with lowest security risk for device j is given by

$$f_*^j(\cdot) = \arg\min_{f_n(\cdot)} \sum_{t=1}^{T} r_t^j\big(f_n(\mathcal{K}_t^j)\big). \tag{14}$$

And in this case, the regret on the average risk is

$$\mathrm{Reg}_T^j := \sum_{t=1}^{T} \mathbb{E}\big[r_t^j(a_t^j)\big] - \sum_{t=1}^{T} r_t^j\big(f_*^j(\mathcal{K}_t^j)\big) \tag{15}$$

where the expectation accounts for the randomness of the algorithm itself. Hence, for device j, if the accumulated security risk of an algorithm is comparable to that incurred by $f_*^j(\cdot)$, which is the best policy in *hindsight*, the algorithm is desirable. The benchmark $f_*^j(\cdot)$ is a stationary policy, while the optimal one can be non-stationary. The benchmark to compare with in (15) boils down to the best server list π_*^j, which is equivalent to finding $\bar{\mathbf{p}}^{j*} = [0, \ldots, 1, \ldots, 0]^\top$, namely

$$r_t^j\big(f_*^j(\mathcal{K}_t^j)\big) = \big(\bar{\mathbf{p}}^{j*}\big)^\top \mathbf{\Gamma}(\mathcal{K}_t^j)\mathbf{r}_t^j = \big(\bar{\mathbf{p}}^{j*}\big)^\top \bar{\mathbf{r}}_t^j. \tag{16}$$

Hence, the regret in (15) can be rewritten as [cf. (16)]

$$\mathrm{Reg}_T^j = \sum_{t=1}^{T} \big(\bar{\mathbf{p}}_t^j\big)^\top \bar{\mathbf{r}}_t^j - \big(\bar{\mathbf{p}}^{j*}\big)^\top \bar{\mathbf{r}}_t^j \tag{17}$$

which will further facilitate the analysis. And the overall regret averaged over all devices is $\mathrm{Reg}_T := (1/J)\sum_{j=1}^{J} \mathrm{Reg}_T^j$.

Our main result relies on the following assumption.

(as1) *The security risk satisfies* $\max_{t,j,k} r_t^j(k) \leq 1$.

Clearly, (as1) implies that security risks are bounded, which is standard in online learning settings [1], [2], [15].

For the subsequent analysis, we will also need an auxiliary variable $q_t^j(n)$ for policy f_n with $f_n\big(\tilde{\mathcal{K}}_t^j\big) \notin \mathcal{S}_t^j$, defined as

$$\bar{q}_t^j(n) = \frac{\bar{p}_t^j(n)}{\mu_t^j + \sum_{m=1}^{\bar{K}} \bar{p}_t^j(m) \mathbb{1}\big(f_m(\tilde{\mathcal{K}}_t^j) = f_n(\tilde{\mathcal{K}}_t^j)\big)}, \quad \forall n : f_n\big(\tilde{\mathcal{K}}_t^j\big) \notin \mathcal{S}_t^j \tag{18a}$$

and for policy f_n with $f_n\big(\tilde{\mathcal{K}}_t^j\big) \in \mathcal{S}_t^j$ as

$$\bar{q}_t^j(n) = \frac{\bar{p}_t^j(n)}{\mu_t^j + 1}, \ \forall n : f_n\big(\tilde{\mathcal{K}}_t^j\big) \in \mathcal{S}_t^j. \tag{18b}$$

Let also $Q_t^j := \sum_{n=1}^{\bar{K}} \bar{q}_t^j(n)$, which depends on $\tilde{\mathcal{K}}_t^j$ according to the definition in (18). The value Q_t^j will play an important role in our regret bound established in the next theorem. As expected intuitively, a larger $|\mathcal{S}_t^j|$ leads to a smaller Q_t^j.

Theorem 1. *If jamming is strategic in the sense that* $\mathcal{K}_t^j$ *is chosen adversarially, the regret of SAVE-A is bounded by*

$$\mathrm{Reg}_T \leq \frac{1}{J} \sum_{j=1}^{J} \sum_{t=1}^{T} \bigg(\mu_t^j + \frac{\eta_t^j}{2}\bigg) Q_t^j + \frac{\ln \bar{K}}{\eta_{T+1}^j}. \tag{19}$$

In addition, Q_t^j *is bounded as*

$$\frac{1}{1+\mu_t^j} \leq Q_t^j \leq \big|\mathcal{K}_t^j \cup \mathcal{S}_t^j\big| - \big|\mathcal{S}_t^j\big| + \mathbb{1}\big(\mathcal{S}_t^j \neq \emptyset\big). \tag{20}$$

Proof. See Appendix B. □

To evaluate the performance gain of cooperation, we revisit the performance without information sharing among devices. Consider Alg. 1 without the cooperation step (line 7). Choosing $\eta_t^j = \sqrt{\ln \bar{K}/(KT)}$ and $\mu_t^j = \eta_t^j/2, \forall t, \forall j$, the regret of Alg. 1 without cooperation is bounded by (cf. Appendix C)

$$\mathrm{Reg}_T^j \leq 2\sqrt{TK \ln \bar{K}} = \mathcal{O}\big(\sqrt{TK^2 \ln K}\big) \tag{21}$$

where the equality follows from Stirling's approximation $\ln \bar{K} = K \ln K - K + \mathcal{O}(\ln K)$. If instead $\eta_t^j = \sqrt{\frac{\ln \bar{K}}{2Kt}}$ and $\mu_t^j = \frac{\eta_t^j}{2}$, the regret of Alg. 1 without cooperation is bounded as

$$\text{Reg}_T \leq 2\sqrt{2TK\ln\bar{K}} = \mathcal{O}\big(\sqrt{TK^2\ln K}\big). \tag{22}$$

According to (21) and (22) both fixed and diminishing η_t^j as well as μ_t^j, guarantee an $\mathcal{O}(\sqrt{TK^2\ln K})$ regret, which matches that of [20]. This demonstrates that SAVE-A could also perform well even without cooperation.

The following corollary establishes a sublinear regret when SAVE-A is employed with cooperation.

Corollary 1. *Selecting* $\eta_t^j = \sqrt{(\ln\bar{K})/\big(K+\sum_{\tau=1}^{t-1}Q_\tau^j\big)}$*, and* $\mu_t^j = \eta_t^j/2, \forall t, j$*, the regret is bounded by*

$$\text{Reg}_T \leq \frac{2}{J}\sum_{j=1}^{J}\sqrt{\sum_{t=1}^{T}Q_t^j\ln\bar{K}}. \tag{23}$$

Proof. See Appendix D. □

The adaptive learning rate η_t^j used in Corollary 1 is still causal, because it does not need information of the current Q_t^j. To assess the benefit of cooperation, we will rely on what we term *value of cooperation* expressed as the ratio of the regret bound in (23) over that in (21), namely

$$\lambda := \frac{\mathbb{E}\Big[\sum_{j=1}^{J}\sqrt{\sum_{t=1}^{T}Q_t^j\ln\bar{K}}\Big]}{J\sqrt{TK\ln\bar{K}}}. \tag{24}$$

The value of cooperation quantifies the improvement of leveraging cooperation among devices. Ideally, $\lambda \leq 1$ suggests that the cooperation reduces the security risk in the worst case. By plugging the bound of Q_t^j in (20) into (23), we arrive at the following corollary that offers an upper bound on λ.

Corollary 2. *The cooperation value of SAVE-A satisfies*

$$\lambda \leq \frac{1}{J}\sum_{j=1}^{J}\sqrt{\frac{1}{T}+\frac{1}{KT}\sum_{t=1}^{T}\Big(\big|\mathcal{K}_t^j\cup\mathcal{S}_t^j\big|-\big|\mathcal{S}_t^j\big|+\mathbb{1}\big(\mathcal{S}_t^j\neq\emptyset\big)\Big)}. \tag{25}$$

The bound in (25) depends on $\mathcal{K}_t^j$ as well as $\mathcal{S}_t^j$. Since $\big|\mathcal{K}_t^j\cup\mathcal{S}_t^j\big|-\big|\mathcal{S}_t^j\big| \leq \big|\mathcal{K}_t^j\big| \leq K$, when T is large enough so that $1/T$ is sufficiently small, we deduce that $\lambda \leq 1$. On the other hand, a larger $\big|\mathcal{S}_t^j\big|$ leads to a smaller λ, suggesting that cooperation indeed helps to secure edge computing. An interesting future research direction is to investigate a lower bound on λ.

IV. Edge Computing under Stochastic Jamming

This section deals with secure edge computing when jamming is modeled stochastically with the probability of each wireless link being jammed following a fixed distribution. The initial version of this scheme can be found in [21], but the analysis of the algorithm and the reason why stochasticity of attacks can simplify the algorithm design are novel.

A. Reduced-complexity stochastic model of jamming

Viewing the sample path of randomly available $\{\mathcal{K}_t^j\}$ as a sequence of deterministic server sets, jamming can be handled using SAVE-A. However, the search space in SAVE-A has size $\big|\bar{\mathcal{K}}\big| = \bar{K}$, which still grows as K! It will be argued next that *a stochastic jamming model can further reduce complexity of secure server selection.*

An attractive feature of random attacks is that the probability of having an available server set $\tilde{\mathcal{K}}$, denoted as $\Pr(\mathcal{K}_t^j = \tilde{\mathcal{K}})$, is fixed over time. Clearly, $\Pr(\mathcal{K}_t^j = \tilde{\mathcal{K}})$ depends on the marginal probability of each server being available. For a given policy f_n, this implies that the expected risk of device j at slot t is

$$\begin{aligned}\mathbb{E}_{\mathcal{K}_t^j}[r_t^j(f_n(\mathcal{K}_t^j))] &= \sum_{\tilde{\mathcal{K}}\in 2^{\mathcal{K}}}\Pr\left(\mathcal{K}_t^j=\tilde{\mathcal{K}}\right)r_t^j\left(f_n(\tilde{\mathcal{K}})\right) \qquad (26)\\ &= \sum_{\tilde{\mathcal{K}}\in 2^{\mathcal{K}}}\Pr\left(\mathcal{K}_t^j=\tilde{\mathcal{K}}\right)\sum_{k=1}^{K}r_t^j(k)\mathbb{1}(f_n(\tilde{\mathcal{K}})=k)\end{aligned}$$

where $2^{\mathcal{K}}$ is the power set of $\mathcal{K}$. Exchanging the order of summations in the RHS of (26), we arrive at

$$\begin{aligned}\mathbb{E}_{\mathcal{K}_t^j}[r_t^j(f_n(\mathcal{K}_t^j))] &= \sum_{k=1}^{K}\Bigg[\sum_{\tilde{\mathcal{K}}\in 2^{\mathcal{K}}}\Pr(\mathcal{K}_t^j=\tilde{\mathcal{K}})\mathbb{1}(f_n(\tilde{\mathcal{K}})=k)\Bigg]r_t^j(k)\\ &= \sum_{k=1}^{K}\mathbb{E}_{\mathcal{K}_t^j}\big[\mathbb{1}(f_n(\mathcal{K}_t^j)=k)\big]r_t^j(k)\\ &= \sum_{k=1}^{K}\Pr\left(f_n(\mathcal{K}_t^j)=k\right)r_t^j(k). \qquad (27)\end{aligned}$$

Therefore, for any pmf $\bar{\mathbf{p}} \in \Delta^{\bar{K}}$ over policies, we have

$$\begin{aligned}\mathbb{E}_{\mathcal{K}_t^j}\Big[\bar{\mathbf{p}}^\top\boldsymbol{\Gamma}(\mathcal{K}_t^j)\mathbf{r}_t^j\Big] &= \sum_{n=1}^{\bar{K}}\bar{p}(n)\sum_{k=1}^{K}\mathbb{P}\big(f_n(\mathcal{K}_t^j)=k\big)r_t^j(k)\\ = \sum_{k=1}^{K}r_t^j(k)&\underbrace{\sum_{n=1}^{\bar{K}}\bar{p}(n)\Pr\big(f_n(\mathcal{K}_t^j)=k\big)}_{=p(k)} = \sum_{k=1}^{K}p(k)r_t^j(k)\end{aligned} \tag{28}$$

where $\mathbf{p} \in \Delta^K$ is the new weight. Therefore, for a given $\bar{\mathbf{p}}^{j*}$ defined in (17), we have the corresponding $\mathbf{p}^{j*}$, and the expected regret can be written as (cf. (15))

$$\begin{aligned}\mathbb{E}_{\mathcal{K}_t^j}\big[\text{Reg}_T^j\big] &= \sum_{t=1}^{T}\mathbb{E}_{a_t^j,\mathcal{K}_t^j}\big[r_t^j(a_t^j)\big]-\sum_{t=1}^{T}\mathbb{E}_{\mathcal{K}_t^j}\big[r_t^j\big(f_*^j(\mathcal{K}_t^j)\big)\big]\\ &= \sum_{t=1}^{T}\mathbb{E}_{a_t^j,\mathcal{K}_t^j}\big[r_t^j(a_t^j)\big]-\sum_{t=1}^{T}\left(\mathbf{p}^{j*}\right)^\top\mathbf{r}_t^j. \qquad (29)\end{aligned}$$

Note that $\mathbf{p}^{j*} \in \Delta^K$, which is a fixed pmf across servers. With a stochastic jamming model, this implies that if minimizing the expected regret is our ultimate goal, searching over the $\bar{K}$-dimensional policy space is tantamount to searching over the K-dimensional server space. Based on this consideration, the subsequent algorithm design views each arm as a server.

B. SAVE-S for edge computing under stochastic jamming

Suppose that per slot t, device j first selects a server a_t^j (as an arm of the MAB) according to $\mathbf{p}_t^j$, e.g., $a_t^j \sim \mathbf{p}_t^j$ and $a_t^j \in \mathcal{K}_t^j$. Once the computation tasks are completed by the end of slot t, device j observes not only the risk of server a_t^j, but thanks to cooperation, also the risk of servers in $\mathcal{S}_t^j$ which can include servers not in $\mathcal{K}_t^j$. Similar to SAVE-A, it is possible

Algorithm 2 SAVE-S for IoT device j

```
1:  Initialize: weight $\mathbf{w}_1^j = \mathbf{1}/K$, exploration factor $\mu_t^j$,
    learning rate $\eta_t^j$.
2:  for $t = 1, 2, \ldots, T$ do
3:      Available server set $\mathcal{K}_t^j$ is revealed.
4:      Update $\mathbf{p}_t^j$ via (32) and choose server $a_t^j \sim \mathbf{p}_t^j$.
5:      Receive $\gamma_{c,t}(a_t^j)$ and $\gamma_{s,t}(a_t^j)$.
6:      Broadcast $\gamma_{c,t}(a_t^j)$ and $\gamma_{s,t}(a_t^j)$ to devices in $\{i \,|\, j \in \mathcal{S}_t^i\}$.
7:      Compute security risk for $\{a_t^j\} \cup \mathcal{S}_t^j$ via (1).
8:      Estimate the security risk via (30).
9:      Update $\mathbf{w}_{t+1}^j$ via (31).
10: end for
```

for device j to observe the risk of servers in $\tilde{\mathcal{K}}_t^j = \mathcal{K}_t^j \cup \mathcal{S}_t^j$. For servers in $\tilde{\mathcal{K}}_t^j \setminus \mathcal{S}_t^j$, the risk is observed only when device j chooses the corresponding server; while the cooperation across devices brings the observable risks for servers in $\mathcal{S}_t^j$. Hence, the risk estimator is constructed via

$$\hat{r}_t^j(k) = \begin{cases} \frac{r_t^j(k)\mathbb{1}\left(a_t^j = k\right)}{\mu_t^j + p_t^j(k)}, & \forall k \in \tilde{\mathcal{K}}_t^j \setminus \mathcal{S}_t^j \\ \frac{r_t^j(k)}{\mu_t^j + 1}, & \forall k \in \mathcal{S}_t^j \\ 0, & \text{else} \end{cases} . \tag{30}$$

Similar to SAVE-A, the presence of μ_t^j in the denominator confines the exploration thanks to the shared risk information. The value of μ_t^j will be specified in Corollary 3.

To estimate the kth server's accumulated risk up to slot t, the following unnormalized weight is maintained per device j

$$w_{t+1}^j(k) = \exp\Big(-\eta_{t+1}^j \sum_{\tau=1}^{t} \hat{r}_\tau^j(k)\Big), \ \forall k \in \mathcal{K} \tag{31}$$

where the time-varying learning rate η_{t+1}^j again accounts for the cooperation. By normalizing (31), device j finds the server selection probability as

$$p_{t+1}^j(k) = \frac{w_{t+1}^j(k)\mathbb{1}\left(k \in \mathcal{K}_{t+1}^j\right)}{\sum_{m \in \mathcal{K}_{t+1}^j} w_{t+1}^j(m)}, \ \forall k \in \mathcal{K} \ . \tag{32}$$

Note that in practice, $p_{t+1}^j(k)$ can only be observed after $\mathcal{K}_{t+1}$ is observed, e.g., at beginning of slot $t+1$. From (32), it is clear that for server $k \notin \mathcal{K}_{t+1}^j$, we have $p_{t+1}^j(k) = 0$. The SAVE-S algorithm is summarized in Alg. 2.

C. Regret analysis for SAVE-S

To establish a regret bound for SAVE-S, consider the auxiliary variable

$$q_t^j(k) = \frac{p_t^j(k)}{\mu_t^j + p_t^j(k)\mathbb{1}(k \notin \mathcal{S}_t^j) + \mathbb{1}(k \in \mathcal{S}_t^j)}, \ \forall k \tag{33}$$

and let also $Q_t^j := \sum_{k=1}^{K} q_t^j(k)$. Depending on $\mathcal{S}_t^j$, the value of Q_t^j is smaller when $|\mathcal{S}_t^j|$ is larger. Then under (as1), the following theorem establishes the desirable regret bound.

Theorem 2. *If the wireless links are stochastically attacked by jammers, the regret of SAVE-S can be bounded by*

$$\mathbb{E}\big[\mathrm{Reg}_T\big] \le \frac{1}{J}\mathbb{E}\left[\sum_{j=1}^{J}\sum_{t=1}^{T}\left(\mu_t^j + \frac{\eta_t^j}{2}\right)Q_t^j + \frac{\ln K}{\eta_{T+1}^j}\right] \tag{34}$$

where the expectation is over the randomness of $\mathcal{K}_t^j$. and the auxiliary variable Q_t^j is bounded by

$$Q_t^j \le \min\left\{K, K+1-|\mathcal{S}_t^j|\right\} \tag{35}$$

Proof. See Appendix E. □

Similar to SAVE-A, the benefit of cooperation here is also assessed by λ [cf. (24)]. To evaluate λ, consider first Alg. 2 without the cooperation step (line 7). By choosing $\eta_t^j = \sqrt{\ln K/(KT)}$ and $\mu_t^j = \eta_t^j/2, \forall t, j$, the regret is

$$\mathbb{E}\big[\mathrm{Reg}_T\big] \le 2\sqrt{TK\ln K}. \tag{36}$$

Diminishing learning rate $\eta_t^j = \sqrt{\ln K/(2Kt)}$ and $\mu_t^j = \eta_t^j/2, \forall t, j$ can be adopted if T is unknown at the beginning of SAVE-S. In this case, the regret is bounded by

$$\mathbb{E}\big[\mathrm{Reg}_T\big] \le 2\sqrt{2TK\ln K}. \tag{37}$$

The derivations are similar to (21) and (22), and thus omitted.

In the ensuing corollary, we present a tighter bound.

Corollary 3. *If IoT devices cooperate, and we choose time-varying learning rates $\eta_t^j = \sqrt{(\ln K)/\big(K + \sum_{\tau=1}^{t-1} Q_\tau^j\big)}$, as well as $\mu_t^j = \eta_t^j/2, \forall t, j$, the regret can be bounded as*

$$\mathbb{E}\big[\mathrm{Reg}_T\big] \le \frac{2}{J}\mathbb{E}\left[\sum_{j=1}^{J}\sqrt{\sum_{t=1}^{T} Q_t^j \ln K}\right]. \tag{38}$$

Proof. It follows steps similar to those Corollary 1. □

The time-varying learning rate η_t^j used in Corollary 3 is still causal because it does not need the current Q_t^j. As $\sum_{t=1}^{T} Q_t^j \le KT$, it follows readily that the bound in (38) is better than those in (36) and (37). The upper bound of the cooperation value λ is given in the corollary next.

Corollary 4. *The cooperation value of SAVE-S satisfies*

$$\lambda \le \frac{1}{J}\sum_{j=1}^{J}\sqrt{\frac{1}{T} + \frac{1}{KT}\sum_{t=1}^{T}\min\left\{K, K+1-|\mathcal{S}_t^j|\right\}}. \tag{39}$$

Proof. See Appendix F. □

The bound in (39) asserts that more side observations will reduce the regret. Specifically, if $K/2$ servers' information can be obtained per device j via information sharing at each slot, e.g., $|\mathcal{S}_t^j| = \frac{K}{2}$, it leads to $\lambda \le \sqrt{\frac{1}{T} + \frac{1}{2} + \frac{1}{K}}$.

V. Simulations

In this section, numerical tests are presented based on both synthetic and real data.

A. Synthetic data tests

Consider $K = 5$ edge servers, and $J = 1$ device with $\rho = 0.8$ in (1), over $T = 400$ slots. Since only one device is considered, we omit the superscript j. The resource c_t required

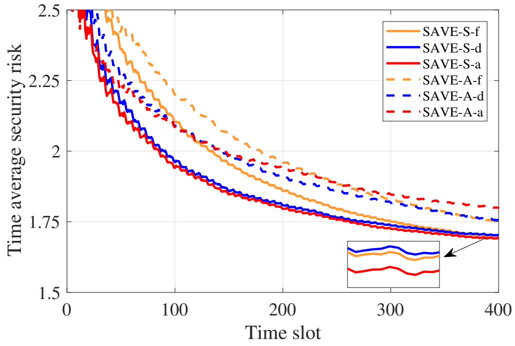


Fig. 1. SAVE-S and SAVE-A without cooperations under stochastic jamming attacks with fixed (f), diminishing (d), and adaptive (a) stepsizes.

TABLE I
SIDE OBSERVATION (SO) PROBABILITY

Server	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
Is SO	1	1	0	0	1	0.3	1	0.6	0.5	0
Not SO	0	0	1	1	0	0.7	0	0.4	0.5	1

for computing is generated as $c_t = (0.6+0.5v_t)\cos 2t$, where v_t is uniformly distributed in $[0,1]$; and s_t is given by $s_t = (0.25+0.3v_t)x_t$, where v_t is again uniformly distributed in $[0,1]$; and x_t is also a uniform random variable over $[0.8,1.2]$. The corresponding security risks $\boldsymbol{\gamma}_{c,t}$ are generated as

$$\gamma_{c,t}(k) = \frac{2k}{3}\big(|\sin t| + 0.8 + |v_1|\big) \tag{40}$$

with v_1 being a Gaussian random variable $v_1 \sim \mathcal{N}(0, 1.44)$; while $\boldsymbol{\gamma}_{s,t}$ is generated as

$$\gamma_{s,t}(k) = \frac{k}{2}\big(0.5\sin t + 0.75 + |v_2|\big) \tag{41}$$

with $v_2 \sim \mathcal{N}(0, 0.64)$.

Before considering side observations, the effectiveness of SAVE-A and SAVE-S is tested without cooperation among devices. For fairness, we consider stochastic jammers, where SAVE-A and SAVE-S both enjoy theoretical guarantees. The jamming probability of edge servers is listed in the left part of Table II. The security risks of SAVE-A and SAVE-S with different learning rates are plotted in Fig. 1. It can be seen that SAVE-S outperforms SAVE-A, since the shrunk search space requires less exploration.

To showcase the improvement attained from side observations, rather than receiving security risks directly from other devices, the side observations are obtained probabilistically. Specifically, for the first 200 slots, the probability of revealing each edge server's risk is listed in the white part of Table I; and for the rest of the slots, it is listed in the shaded part of Table I. For comparison, we consider two schemes: i) round robin: cyclically choose the server from the available ones; ii) random selection, where we randomly choose the server from the available ones.

No jamming. In Figs. 2 (a1) and (a2), the SAVE-S and SAVE-A are compared with their corresponding non-cooperative counterparts. Both SAVE-S and SAVE-A outperform the round robin and random selection significantly. Clearly, the cooperation improves the *time-average security risk* of SAVE-S by a percentage of 10.02%, 17.39%, and 17.42% for fixed, diminishing, and adaptive learning rates, respectively. Regarding SAVE-A, the improvement thanks to cooperation is 19.07%, 19.42% and 20.12% with fixed, diminishing, and adaptive learning rates. As confirmed by simulations, cooperation improves the regret performance of SAVE-S and SAVE-A considerably; e.g., the cooperation values are $\lambda = 0.51$ and 0.50, respectively.

TABLE II
SERVER ON/OFF PROBABILITY

Server	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
On	0.7	0.8	0.9	1	0.6	0.3	1	0.6	0.5	0.8
Off	0.3	0.2	0.1	0	0.4	0.7	0	0.4	0.5	0.2

TABLE III
SERVER ON/OFF PROBABILITY

Link	1 to 2	1 to 3	2 to 1	2 to 3	3 to 1	3 to 2
Cooperation	0.1	0.4	0	0.5	0.6	0.3
No Cooperation	0.9	0.6	1	0.5	0.4	0.7

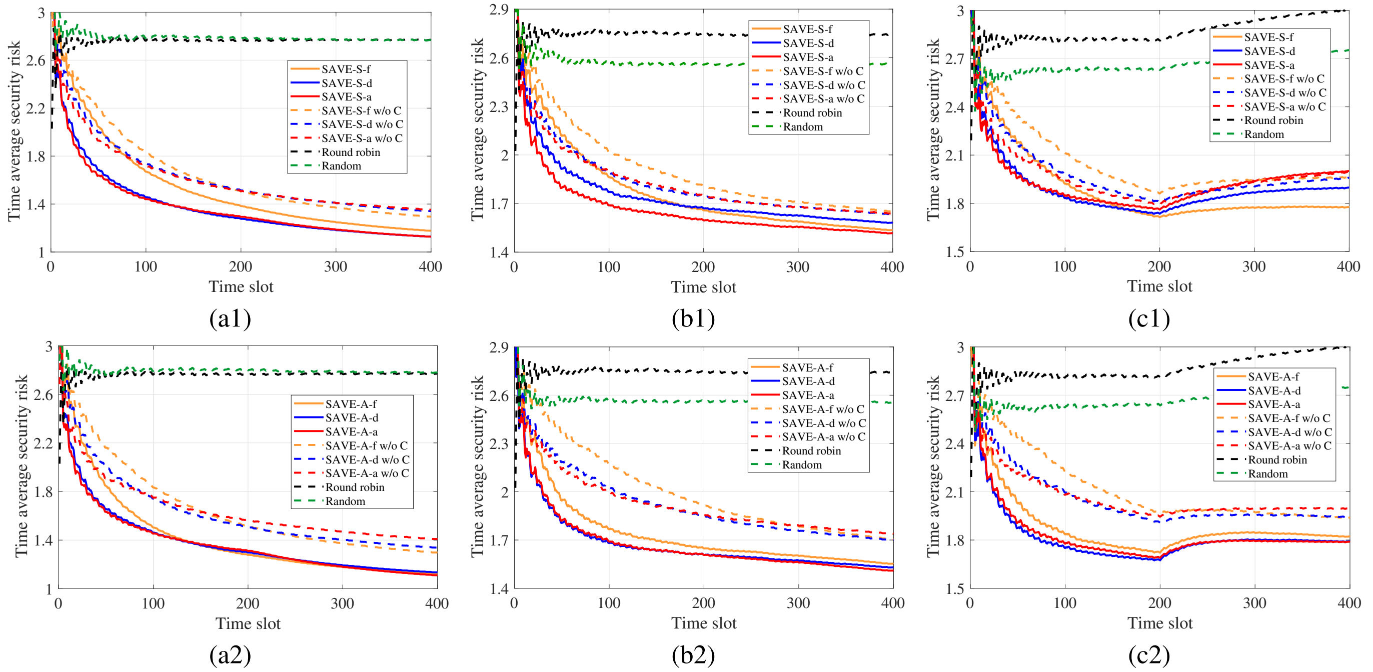
Stochastic jamming attacks. Suppose that the servers are under attack by stochastic jammers, where the on-off probability of edge servers is listed in Table II. The simulations shown in Figs. 2 (b1) and (b2) illustrate that cooperation improves the time-average security risk of SAVE-S by 6.08%, 3.37% and 6.13% under fixed, diminishing, and adaptive learning rates, respectively. In this test, the cooperation value is $\lambda = 0.50$. Regarding SAVE-A, the improvement provided by cooperation is 8.83%, 10.12%, and 13.21% when fixed, diminishing, and adaptive learning rates are adopted, where the cooperation value is $\lambda = 0.51$.

Adversarial jamming attacks. With adversarial jammers, the difference in data generation is that in the first 200 slots the probability of server being jammed follows the left part of Table II, while the rest of the time the probability follows the right part of Table II. Fig. 2 (c1) depicts the performance of SAVE-S, which is not guaranteed to attain sublinear regret. This also explains why the risk-order list of servers is necessary for algorithms in adversarial jamming attacks. On the other hand, Fig. 2 (c2) compares the performance of SAVE-A with different learning rates. The cooperation improves the time-average security risk of SAVE-A by 7.14%, 8.28% and 10.26% for fixed, diminishing, and adaptive learning rates respectively, along with $\lambda = 0.54$.

B. Real data tests

The performance of SAVE-S and SAVE-A is further tested on a real world dataset [26], which contains the customers' feedback on cloud services from public websites such as Cloud Hosting Reviews, where more than 10,000 pieces of information feedback from nearly 7,000 consumers over 113 cloud services are collected. The consumers' feedback is the service trust (using risk $\boldsymbol{\gamma}_{c,t}$ and $\boldsymbol{\gamma}_{s,t}$ for negative trust). In this test, we consider $K = 3$ edge servers and $J = 3$ IoT devices. The information sharing probabilities are listed in Table III.

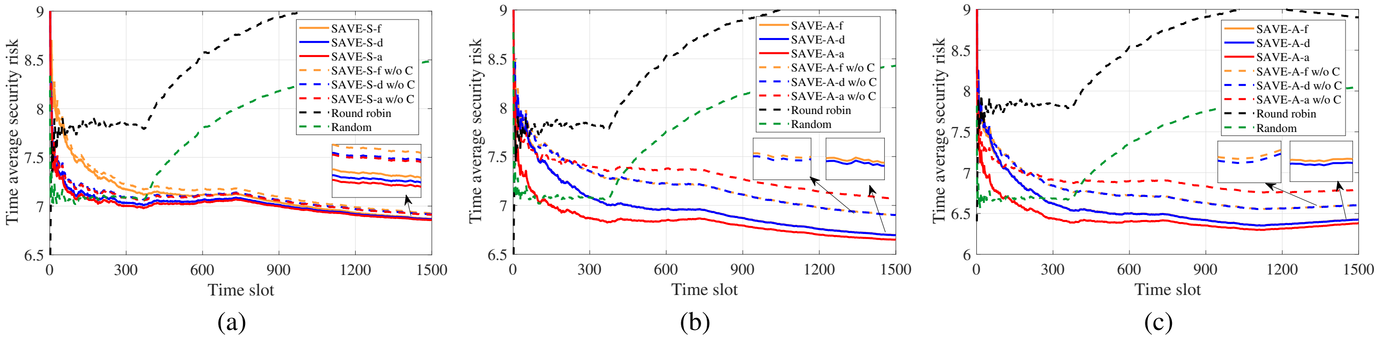
How SAVE-S performs in combating stochastic jammers is shown in Fig. 3 (a). SAVE-S with different learning rates outperforms round robin and randomized server selection schemes. In this case, cooperation slightly improves the time-average security risk by 1.82%, 2.21% and 2.64% for fixed, diminishing, and adaptive learning rates. The cooperation value in this case is $\lambda = 0.71$. This suggests that if the "worst case"



(a1) (b1) (c1)

(a2) (b2) (c2)

Fig. 2. Synthetic data tests: (a1) SAVE-S without jamming attacks; (a2) SAVE-A without jamming attacks; (b1) SAVE-S with stochastic jamming attacks; (b2) SAVE-A with stochastic jamming attacks; (c1) SAVE-S with adversarial jamming attacks; (c2) SAVE-A with adversarial attacks.



(a) (b) (c)

Fig. 3. Real data tests: (a) SAVE-S for stochastic jamming attacks; (b) SAVE-A for stochastic attacks; (c) SAVE-A for adversarial jamming attacks.

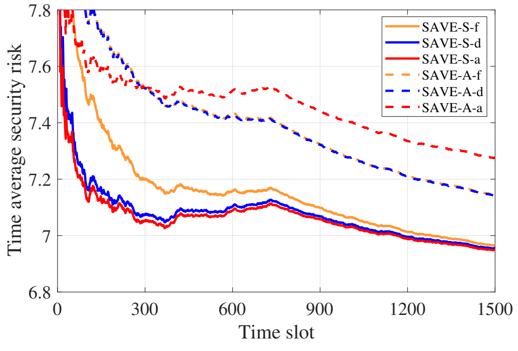


Fig. 4. A comparison of SAVE-S and SAVE-A using real data.

(in terms of $\mathcal{K}_t^j$ selection) does not occur, the performance of SAVE-S is good enough. The performance of SAVE-A under stochastic jamming is shown in Fig. 3 (b). In this case, the cooperation improves the time-average security risk by 4.14%, 4.23% and 5.36% for fixed, diminishing, and adaptive learning rates respectively, together with a cooperation value $\lambda = 0.63$.

Regarding adversarial jammers, Fig. 3 (c) shows that cooperation improves the time-average security risk of SAVE-A by 4.32%, 4.48% and 5.22% for fixed, diminishing, and adaptive learning rates, respectively, while the cooperation value is $\lambda = 0.63$. We further compare the time-averaged risks in Fig. 4 in the presence of stochastic jamming attacks without cooperation. It is seen that SAVE-S with different learning rates outperforms SAVE-A, which again demonstrates the benefit of the reduced-size search space in SAVE-S.

VI. Conclusions

Online security-aware edge computing under jamming attacks was studied in this paper. Different from increasing bandwidth or transmission power, we introduced schemes suitable for low-power IoT devices. Specifically, we developed two algorithms to offload tasks to the most reliable server when adversarial and stochastic jamming attacks are present, respectively. Sublinear regret for both schemes was analytically established. Performance of the proposed algorithms was further enhanced via cooperation among devices. Analysis confirmed the value of cooperation through a considerable improvement on the regret bound. Numerical tests on both synthetic and real datasets demonstrated the effectiveness of the proposed schemes.

APPENDIX

A. *Proof of Lemma 1*

To show that $\big(\bar{\mathbf{p}}_t^j\big)^\top \bar{\mathbf{r}}_t^j = \big(\mathbf{p}_t^j\big)^\top \mathbf{r}_t^j$, it suffices to prove that given $\mathcal{K}_t^j$ and $\mathbf{p}_t^j \in \Delta(\mathcal{K}_t^j)$, it holds that $\big(\bar{\mathbf{p}}_t^j\big)^\top \mathbf{\Gamma}\big(\mathcal{K}_t^j\big) = \big(\mathbf{p}_t^j\big)^\top$. To this end, we will use the special structure of $\mathbf{\Gamma}\big(\mathcal{K}_t^j\big) \in \{0,1\}^{\bar{K}\times K}$. For $k \notin \mathcal{K}_t^j$, all entries of the k-th column of $\mathbf{\Gamma}\big(\mathcal{K}_t^j\big)$ are 0. And for $k \in \mathcal{K}_t^j$, the k-th column has $\bar{K}/K$ entries equal to 1 and the rest 0. Besides, each row of $\mathbf{\Gamma}\big(\mathcal{K}_t^j\big)$ has only one non-zero entry because each risk-ordered list has one policy outcome given $\mathcal{K}_t^j$. Without loss of generality, let rows $(m-1)\bar{K}/K+1, \dots, \bar{K}/K$ of $\mathbf{\Gamma}\big(\mathcal{K}_t^j\big)$ be of the form $[0,\dots,1,\dots,0]$ with the m-th entry being 1. Then $\big(\bar{\mathbf{p}}_t^j\big)^\top \mathbf{\Gamma}\big(\mathcal{K}_t^j\big) = \big(\mathbf{p}_t^j\big)^\top$ becomes $\sum_{k=(m-1)\bar{K}/K+1}^{m\bar{K}/K} \bar{p}_t^j(k) = p_t^j(m), \forall m \in \mathcal{K}$, which has at least one solution in $\Delta^{\bar{K}}$.

B. *Proof of Theorem 1*

To start, let us introduce the cumulative risk estimate

$$\hat{\bar{R}}_t^j(n) := \sum_{\tau=1}^{t} \hat{\bar{r}}_\tau^j(n),\ \forall n \in \bar{\mathcal{K}}. \tag{42}$$

Upon defining auxiliary variables $W_t^j := \sum_{n=1}^{\bar{K}} \exp\big[-\eta_t^j \hat{\bar{R}}_{t-1}^j(n)\big]$, and $\tilde{W}_t^j := \sum_{n=1}^{\bar{K}} \exp\big[-\eta_{t-1}^j \hat{\bar{R}}_{t-1}^j(n)\big]$, we have

$$\begin{aligned}\frac{1}{\eta_t^j}\ln\left(\frac{\tilde{W}_{t+1}^j}{W_t^j}\right) &= \frac{1}{\eta_t^j}\ln\left(\frac{\sum_{n=1}^{\bar{K}}\exp\big[-\eta_t^j\hat{\bar{R}}_t^j(n)\big]}{W_t^j}\right)\\ &= \frac{1}{\eta_t^j}\ln\left(\sum_{n=1}^{\bar{K}}\frac{\bar{w}_t^j(n)\exp\big[-\eta_t^j\hat{\bar{r}}_t^j(n)\big]}{W_t^j}\right)\\ &= \frac{1}{\eta_t^j}\ln\left(\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\exp\big[-\eta_t^j\hat{\bar{r}}_t^j(n)\big]\right)\\ &\overset{(a)}{\leq} \frac{1}{\eta_t^j}\ln\left(\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\Big(1-\eta_t^j\hat{\bar{r}}_t^j(n)+\frac{1}{2}\big(\eta_t^j\hat{\bar{r}}_t^j(n)\big)^2\Big)\right).\end{aligned} \tag{43}$$

The first equality is due to

$$\bar{w}_t^j(n) = \exp\left(-\eta_t^j\sum_{\tau=1}^{t-1}\hat{\bar{r}}_\tau^j(n)\right) = \exp\left(-\eta_t^j\hat{\bar{R}}_{t-1}^j(n)\right)$$

which further implies

$$\begin{aligned}\exp\left(-\eta_t^j\hat{\bar{R}}_t^j(n)\right) &= \exp\left(-\eta_t^j\Big[\hat{\bar{R}}_{t-1}^j(n)+\hat{\bar{r}}_t^j(n)\Big]\right)\\ &= \bar{w}_t^j(n)\exp\big(-\eta_t^j\hat{\bar{r}}_t^j(n)\big).\end{aligned}$$

And (a) in (43) is due to $e^{-x} \leq 1-x+\frac{x^2}{2},\ \forall x \geq 0$.

The upper bound in (43) can be further bounded as

$$\begin{aligned}&\frac{1}{\eta_t^j}\ln\left(\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\Big(1-\eta_t^j\hat{\bar{r}}_t^j(n)+\frac{1}{2}\big(\eta_t^j\hat{\bar{r}}_t^j(n)\big)^2\Big)\right)\\ &= \frac{1}{\eta_t^j}\ln\left(1-\eta_t^j\sum_{n=1}^{\bar{K}}\hat{\bar{r}}_t^j(n)\bar{p}_t^j(n)+\frac{\big(\eta_t^j\big)^2}{2}\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\big(\hat{\bar{r}}_t^j(n)\big)^2\right)\\ &\overset{(b)}{\leq} -\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\hat{\bar{r}}_t^j(n)+\frac{\eta_t^j}{2}\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\big(\hat{\bar{r}}_t^j(n)\big)^2\end{aligned} \tag{44}$$

where (b) uses $\ln(1-x) \leq -x,\ \forall x \geq 0$. Therefore, we have

$$\frac{1}{\eta_t^j}\ln\left(\frac{\tilde{W}_{t+1}^j}{W_t^j}\right) \leq -\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\hat{\bar{r}}_t^j(n)+\frac{\eta_t^j}{2}\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\big(\hat{\bar{r}}_t^j(n)\big)^2. \tag{45}$$

Rearranging (45), we arrive at

$$\begin{aligned}\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\hat{\bar{r}}_t^j(n) &\leq \frac{\eta_t^j}{2}\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\big(\hat{\bar{r}}_t^j(n)\big)^2+\frac{1}{\eta_t^j}\ln\frac{W_t^j}{\tilde{W}_{t+1}^j}\\ &= \frac{\eta_t^j}{2}\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\big(\hat{\bar{r}}_t^j(n)\big)^2\\ &\quad+\left(\frac{\ln W_t^j}{\eta_t^j}-\frac{\ln W_{t+1}^j}{\eta_{t+1}^j}\right)+\left(\frac{\ln W_{t+1}^j}{\eta_{t+1}^j}-\frac{\ln \tilde{W}_{t+1}^j}{\eta_t^j}\right).\end{aligned} \tag{46}$$

To bound $\frac{\ln W_{t+1}^j}{\eta_{t+1}^j}-\frac{\ln \tilde{W}_{t+1}^j}{\eta_t^j}$, notice that

$$\begin{aligned}W_{t+1}^j &= \sum_{n=1}^{\bar{K}}\exp\big[-\eta_{t+1}^j\hat{\bar{r}}_t^j(n)\big]\\ &\overset{(c)}{\leq} \bar{K}\left(\sum_{n=1}^{\bar{K}}\frac{1}{\bar{K}}\exp[-\eta_t^j\hat{\bar{r}}_t^j(n)]\right)^{\eta_{t+1}^j/\eta_t^j}\\ &= \bar{K}^{(\eta_t^j-\eta_{t+1}^j)/\eta_t^j}\left(\sum_{n=1}^{\bar{K}}\exp\big[-\eta_t^j\hat{\bar{r}}_t^j(n)\big]\right)^{\eta_{t+1}^j/\eta_t^j}\\ &= \bar{K}^{(\eta_t^j-\eta_{t+1}^j)/\eta_t^j}\big(\tilde{W}_{t+1}^j\big)^{\eta_{t+1}^j/\eta_t^j}\end{aligned} \tag{47}$$

where (c) is from $\eta_{t+1}^j \leq \eta_t^j$, and the concavity of $(\cdot)^{\eta_{t+1}^j/\eta_t^j}$. Taking logarithms on both sides of (47), and rearranging terms leads to

$$\frac{\ln W_{t+1}^j}{\eta_{t+1}^j}-\frac{\ln \tilde{W}_{t+1}^j}{\eta_t^j} \leq \left(\frac{1}{\eta_{t+1}^j}-\frac{1}{\eta_t^j}\right)\ln\bar{K}. \tag{48}$$

Plugging (48) into (46) and summing up over t, we have

$$\begin{aligned}\sum_{t=1}^{T}\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\hat{\bar{r}}_t^j(n) &\leq \sum_{t=1}^{T}\frac{\eta_t^j}{2}\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\big(\hat{\bar{r}}_t^j(n)\big)^2\\ &\quad+\left(\frac{1}{\eta_{T+1}^j}-\frac{1}{\eta_1^j}\right)\ln\bar{K}+\frac{\ln W_1^j}{\eta_1^j}-\frac{\ln W_{T+1}^j}{\eta_{T+1}^j}\\ &\overset{(d)}{=} \sum_{t=1}^{T}\frac{\eta_t^j}{2}\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\big(\hat{\bar{r}}_t^j(n)\big)^2+\left(\frac{1}{\eta_{T+1}^j}-\frac{1}{\eta_1^j}\right)\ln\bar{K}-\frac{\ln W_{T+1}^j}{\eta_{T+1}^j}\end{aligned} \tag{49}$$

where (d) follows from $W_1^j = 1$.

Defining $\tilde{p}_t^j(n) = 1$, for $n : f_n\big(\tilde{\mathcal{K}}_t^j\big) \in \mathcal{S}_t^j$, and $\tilde{p}_t^j(n) = \sum_{m=1}^{\bar{K}}\bar{p}_t^j(m)\mathbb{1}\big(f_m(\tilde{\mathcal{K}}_t^j) = f_n(\tilde{\mathcal{K}}_t^j)\big)$, for $n : f_n\big(\tilde{\mathcal{K}}_t^j\big) \notin \mathcal{S}_t^j$, we have that

$$\begin{aligned}\mathbb{E}\left[\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\hat{\bar{r}}_t^j(n)\right] &= \sum_{k=1}^{\bar{K}}\bar{p}_t^j(n)\bar{r}_t^j(n)\left(1-\frac{\mu_t^j}{\mu_t^j+\tilde{p}_t^j(n)}\right)\\ &= \sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\bar{r}_t^j(n)-\sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\bar{r}_t^j(n)\frac{\mu_t^j}{\mu_t^j+\tilde{p}_t^j(n)}\\ &\overset{(e)}{\geq} \sum_{n=1}^{\bar{K}}\bar{p}_t^j(n)\bar{r}_t^j(n)-\mu_t^jQ_t^j\end{aligned} \tag{50}$$

where $\mathbb{E}$ is w.r.t. the probability that $\bar{r}_t^j(n)$ is observed; (e) follows from the assumption $\bar{r}_t^j(n) \leq 1$ and the definition of Q_t^j. The mean-square of the LHS in (49) can be bounded as

$$\mathbb{E}\Big[\sum_{n=1}^{\bar{K}} \bar{p}_t^j(n)\big(\hat{\bar{r}}_t^j(n)\big)^2\Big] \leq \sum_{n=1}^{\bar{K}} \bar{p}_t^j(n)\tilde{p}_t^j(n)\frac{\big(\bar{r}_t^j(n)\big)^2}{\big(\mu_t^j+\tilde{p}_t^j(n)\big)^2}$$
$$\leq \sum_{n=1}^{\bar{K}} \bar{p}_t^j(n)\frac{\big(\bar{r}_t^j(n)\big)^2}{\mu_t^j+\tilde{p}_t^j(n)} \leq \sum_{n=1}^{\bar{K}} \bar{p}_t^j(n)\frac{1}{\mu_t^j+\tilde{p}_t^j(n)} \leq Q_t^j. \quad (51)$$

The mean-square of the third term in the RHS of (49), is

$$\mathbb{E}\Big[-\frac{\ln W_{T+1}^j}{\eta_{T+1}^j}\Big] = \mathbb{E}\Big[-\frac{\ln \sum_{n=1}^{\bar{K}} \bar{w}_{T+1}^j(n)}{\eta_{T+1}^j}\Big]$$
$$\overset{(f)}{\leq} \mathbb{E}\Big[-\frac{\ln \sum_{n=1}^{\bar{K}} \bar{p}^j(n)\bar{w}_{T+1}^j(n)}{\eta_{T+1}^j}\Big] \overset{(g)}{\leq} \mathbb{E}\Big[-\sum_{n=1}^{\bar{K}} \bar{p}^j(n)\frac{\ln \bar{w}_{T+1}^j(n)}{\eta_{T+1}^j}\Big]$$
$$= \mathbb{E}\Big[-\sum_{n=1}^{\bar{K}} \bar{p}^j(n)\frac{-\eta_{T+1}^j\hat{\bar{R}}_{T+1}^j(n)}{\eta_{T+1}^j}\Big] \overset{(h)}{\leq} \sum_{n=1}^{\bar{K}} \bar{p}^j(n)\sum_{t=1}^{T}\bar{r}_t^j(n) \quad (52)$$

where (f) follows since $\ln(\cdot)$ is monotonically increasing and $p^j(k)$ is a fixed distribution; (g) is due to Jensen's inequality; and (h) follows since $\hat{r}_t^j(n)$ and $\hat{R}_t^j(n)$ are under estimators. Taking expectation on (49), and combining (50)-(52), we have

$$\sum_{t=1}^{T}\sum_{n=1}^{\bar{K}} \bar{p}_t^j(n)\bar{r}_t^j(n) - \sum_{t=1}^{T}\sum_{n=1}^{\bar{K}} \bar{p}^j(n)\bar{r}_t^j(n)$$
$$\leq \sum_{t=1}^{T}\Big(\mu_t^j + \frac{\eta_t^j}{2}\Big)Q_t^j + \Big(\frac{1}{\eta_{T+1}^j} - \frac{1}{\eta_1^j}\Big)\ln \bar{K} \quad (53)$$

which completes the proof.

To see the lower bound on $\mathbf{Q}_t^j$

To bound Q_t^j, recall that $Q_t^j = \sum_{n=1}^{\bar{K}} \bar{q}_t^j(n)$ for adversarial jammers, where $\bar{q}_t^j(n)$ is as in (18). Collect the policy indicies n satisfying $f_n(\tilde{\mathcal{K}}_t^j) \in \mathcal{S}_t^j$ in the set $\mathcal{N}_1$; and the remaining policy indices in $\mathcal{N}_2$. For $n \in \mathcal{N}_1$, we then have $\bar{q}_t^j(n) = \bar{p}_t^j(n)/(1+\mu_t^j)$. Since Q_t^j depends on $\mathcal{K}_t^j$ and the side observation set $\mathcal{S}_t^j$, the next lemma specifies their relation.

Lemma 2. *The auxiliary variable Q_t^j is bounded as*

$$Q_t^j \leq \big|\mathcal{K}_t^j \cup \mathcal{S}_t^j\big| - \big|\mathcal{S}_t^j\big| + \mathbb{1}\big(\mathcal{S}_t^j \neq \emptyset\big).$$

Proof. To derive the upper bound, consider first that there are no side observations, meaning $\mathcal{N}_1 = \emptyset$. Then we have

$$Q_t^j = \sum_{k\in\mathcal{K}_t^j}\sum_{n:f_n(\mathcal{K}_t^j)=k} \bar{q}_t^j(n)$$
$$= \sum_{k\in\mathcal{K}_t^j} \frac{\sum_{n:f_n(\mathcal{K}_t^j)=k} \bar{p}_t^j(n)}{\mu_t^j + \sum_{m:f_m(\mathcal{K}_t^j)=k} \bar{p}_t^j(m)} \leq |\mathcal{K}_t^j|. \quad (54)$$

Consider next $\mathcal{N}_1 \neq \emptyset$, which means that side observations are available. For the policies whose indices are collected in $\mathcal{N}_1$, we have

$$\sum_{n\in\mathcal{N}_1} \bar{q}_t^j(n) = \sum_{n\in\mathcal{N}_1} \frac{\bar{p}_t^j(n)}{\mu_t^j+1} \overset{(a)}{\leq} 1, \quad (55)$$

where (a) uses the fact that $\sum_{n\in\mathcal{N}_1} \bar{p}_t^j(n) \leq 1$. Then for policies with indices in $\mathcal{N}_2$, we find

$$\sum_{n\in\mathcal{N}_2} \bar{q}_t^j(n) = \sum_{k\in\tilde{\mathcal{K}}_t^j\setminus\mathcal{S}_t^j}\ \sum_{n:f_n(\tilde{\mathcal{K}}_t^j)=k} \bar{q}_t^j(n)$$
$$= \sum_{k\in\tilde{\mathcal{K}}_t^j\setminus\mathcal{S}_t^j} \frac{\sum_{n:f_n(\tilde{\mathcal{K}}_t^j)=k} \bar{p}_t^j(n)}{\mu_t^j + \sum_{m:f_m(\tilde{\mathcal{K}}_t^j)=k} \bar{p}_t^j(m)} \leq \big|\mathcal{K}_t^j \cup \mathcal{S}_t^j\big| - \big|\mathcal{S}_t^j\big|. \quad (56)$$

Adding (55) and (56), we obtain the upper bound on Q_t^j for $\mathcal{N}_1 \neq \emptyset$. Writing the upper bound on Q_t^j for both $\mathcal{N}_1 \neq \emptyset$ and $\mathcal{N}_1 = \emptyset$ compactly, completes the proof of the lemma. □

Simply plugging the results of Lemma 2 into the definition of λ, we can upper bound λ of SAVE-A as

$$\lambda \leq \frac{1}{J}\sum_{j=1}^{J}\sqrt{\frac{1}{T} + \frac{1}{KT}\sum_{t=1}^{T}\Big(\big|\mathcal{K}_t^j \cup \mathcal{S}_t^j\big| - \big|\mathcal{S}_t^j\big| + \mathbb{1}\big(\mathcal{S}_t^j \neq \emptyset\big)\Big)}. \quad (57)$$

The proof is then complete.

To lower bound Q_t^j, we have by definition that

$$Q_t^j = \sum_{n=1}^{\bar{K}} \bar{q}_t^j(n) \geq \sum_{n=1}^{\bar{K}} \frac{\bar{p}_t^j(n)}{\mu_t^j+1} = \frac{1}{\mu_t^j+1}. \quad (58)$$

C. Derivations of (21) and (22)

Without cooperation, borrowing the result form Lemma 2, and setting $\mathcal{S}_t^j = \emptyset$, we deduce that

$$Q_t^j \leq \big|\mathcal{K}_t^j\big| \leq K. \quad (59)$$

For $\eta_t^j = \sqrt{\frac{\ln \bar{K}}{KT}}$ and $\mu_t^j = \frac{\eta_t^j}{2}$, (19) becomes

$$\text{Reg}_T^j \leq \eta_{T+1}^j KT + \frac{\ln \bar{K}}{\eta_{T+1}^j} = 2\sqrt{TK\ln \bar{K}}. \quad (60)$$

On the other hand, if $\eta_t^j = \sqrt{\frac{\ln \bar{K}}{2Kt}}$ and $\mu_t^j = \frac{\eta_t^j}{2}, \forall t$, the independence between η_t^j and Q_t^j implies

$$\text{Reg}_T^j \leq K\sum_{t=1}^{T}\eta_t^j + \frac{\ln \bar{K}}{\eta_{T+1}^j} \leq 2\sqrt{2TK\ln \bar{K}} \quad (61)$$

where the inequality follows since $\sum_{t=1}^{T} 1/\sqrt{t} \leq 2\sqrt{T}$.

D. Proof of Corollary 1

The proof builds on the following lemma

Lemma 3. *With $Q_1, Q_2, \ldots, Q_T$ and K denoting positive real numbers, the following inequality holds*

$$\sum_{t=1}^{T}\frac{Q_t}{2\sqrt{\delta + \sum_{\tau=1}^{t} Q_\tau}} \leq \sqrt{\delta + \sum_{t=1}^{T} Q_t} - \sqrt{\delta}. \quad (62)$$

Proof. For $x \leq 1$, we have the inequality $\frac{x}{2} \leq 1 - \sqrt{1-x}$. Replacing x with $Q_t/\big(\delta + \sum_{\tau=1}^{t} Q_\tau\big) \leq 1$, we find

$$\frac{Q_t}{2\big(\delta + \sum_{\tau=1}^{t} Q_\tau\big)} \leq 1 - \sqrt{1 - \frac{Q_t}{\delta + \sum_{\tau=1}^{t} Q_\tau}}. \quad (63)$$

Then multiplying both sides with $\sqrt{\delta+\sum_{\tau=1}^{t} Q_\tau}$, we arrive at

$$\frac{Q_t}{2\sqrt{\delta+\sum_{\tau=1}^{t} Q_\tau}} \leq \sqrt{\delta+\sum_{\tau=1}^{t} Q_\tau} - \sqrt{\delta+\sum_{\tau=1}^{t-1} Q_\tau}. \quad (64)$$

Taking summation over T, completes the proof. □

We are now ready to prove the corollary. For a specific realization of Q_t^j, upon choosing $\eta_t^j = \sqrt{(\ln \bar{K})/\big(K+\sum_{\tau=1}^{t-1} Q_\tau^j\big)}$, and $\mu_t^j = \eta_t^j/2$, we arrive at

$$\begin{aligned}
\sum_{t=1}^{T}\left(\mu_t^j + \frac{\eta_t^j}{2}\right)Q_t^j &= \sum_{t=1}^{T} \frac{Q_t^j\sqrt{\ln \bar{K}}}{\sqrt{K+\sum_{\tau=1}^{t-1} Q_\tau^j}} \\
&\leq \sum_{t=1}^{T} \frac{Q_t^j\sqrt{\ln \bar{K}}}{\sqrt{K - Q_t^j + \sum_{\tau=1}^{t} Q_\tau^j}} \overset{(a)}{\leq} \sum_{t=1}^{T} \frac{Q_t^j\sqrt{\ln \bar{K}}}{\sqrt{\delta + \sum_{\tau=1}^{t} Q_\tau^j}} \\
&\overset{(b)}{\leq} \sqrt{\left(\delta + \sum_{t=1}^{T} Q_t\right)\ln \bar{K}} \qquad (65)
\end{aligned}$$

where (a) uses $\delta := \min_{t,j}\{K - Q_t^j\}$ which is strictly greater than 0 according to Lemma 2; and (b) follows from Lemma 3. Then, it is easy to see that

$$\sum_{t=1}^{T}\left(\mu_t^j + \frac{\eta_t^j}{2}\right)Q_t^j + \frac{\ln \bar{K}}{\eta_{T+1}^j} \leq 2\sqrt{\left(\delta + \sum_{t=1}^{T} Q_t\right)\ln \bar{K}} \quad (66)$$

which completes the proof. The bound in (66) can be approximated by $2\sqrt{\sum_{t=1}^{T} Q_t \ln \bar{K}}$, since $\delta \ln \bar{K}$ is not the dominant term.

E. Proof of Theorem 2

The proof starts with a simple case, where for a single device j, we have $\mathcal{K}_t^j = \tilde{\mathcal{K}},\ \forall t$.

Lemma 4. *If $\mathcal{K}_t^j = \tilde{\mathcal{K}},\ \forall t$, then SAVE-S guarantees that*

$$\sum_{t=1}^{T}\sum_{k=1}^{K} p_t^j(k) r_t^j(k) - \sum_{t=1}^{T} r_t^j(k^*) \leq \sum_{t=1}^{T}\left(\mu_t^j + \frac{\eta_t^j}{2}\right)Q_t^j + \frac{\ln K}{\eta_{T+1}^j} \quad (67)$$

where k^* denotes the best fixed server among $\tilde{\mathcal{K}}$ in hindsight.

Proof. It follows steps similar to the proof of Theorem 1. □

Lemma 4 bounds the regret when the active server set is time-invariant. Similar to [19], with the instantaneous regret of device j defined as $V_t^j(\mathcal{K}_t) := \sum_{k=1}^{K} p_t^j(k) r_t^j(k) - r_t\big(f_*^j(\mathcal{K}_t^j)\big)$, the key step is to decouple the regret as

$$\begin{aligned}
\sum_{t=1}^{T}\mathbb{E}\Big[V_t^j(\mathcal{K}_t)\Big] &= \sum_{t=1}^{T}\sum_{\tilde{\mathcal{K}}\subseteq\mathcal{K}} \Pr(\mathcal{K}_t^j = \tilde{\mathcal{K}})\mathbb{E}\Big[V_t^j(\mathcal{K}_t)\big|\mathcal{K}_t^j = \tilde{\mathcal{K}}\Big] \\
&= \sum_{\tilde{\mathcal{K}}\subseteq\mathcal{K}} \Pr(\tilde{\mathcal{K}}) \sum_{t=1}^{T}\mathbb{E}\Big[V_t^j(\tilde{\mathcal{K}})\big|\mathcal{K}_t^j = \tilde{\mathcal{K}}\Big] \\
&\overset{(a)}{=} \sum_{\tilde{\mathcal{K}}\subseteq\mathcal{K}} \Pr(\tilde{\mathcal{K}})\mathbb{E}\left[\sum_{t=1}^{T}\left(\mu_t^j + \frac{\eta_t^j}{2}\right)Q_t^j + \frac{\ln K}{\eta_{T+1}^j}\Bigg|\mathcal{K}_t^j = \tilde{\mathcal{K}}\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\left(\mu_t^j + \frac{\eta_t^j}{2}\right)Q_t^j + \frac{\ln K}{\eta_{T+1}^j}\right] \qquad (68)
\end{aligned}$$

where in (a) we used the result of Lemma 4.

Recall that for stochastic jamming, Q_t^j is defined in (33) as

$$Q_t^j = \sum_{k=1}^{K} q_t^j(k) = \sum_{k=1}^{K} \frac{p_t^j(k)}{\mu_t^j + p_t^j(k)\mathbb{1}(k \notin \mathcal{S}_t^j) + \mathbb{1}(k \in \mathcal{S}_t^j)}. \quad (69)$$

To evaluate the regret bound in Theorem 2, we provide a bound on Q_t^j in the following lemma.

Lemma 5. *If $\mu_t^j \leq 1$ for every t, then Q_t^j is bounded by*

$$\frac{1}{1+\mu_t^j} \leq Q_t^j \leq K - |\mathcal{S}_t^j| + \sum_{k\in\mathcal{S}_t^j} p_t^j(k) - \sum_{k\in\mathcal{S}_t^j} \frac{\mu_t^j p_t^j(k)}{2}. \quad (70)$$

Proof. First, we readily find that

$$Q_t^j = \sum_{k=1}^{K} q_t^j(k) \geq \sum_{k=1}^{K} \frac{p_t^j(k)}{\mu_t^j + 1} = \frac{1}{1+\mu_t^j}. \quad (71)$$

On the other hand, for $k \in \mathcal{S}_t^j$, it holds that $q_t^j(k) = p_t^j(k)/(1+\mu_t^j)$; while for $k \notin \mathcal{S}_t^j$, we have $q_t^j(k) = p_t^j(k)/(p_t^j(k)+\mu_t^j)$. Hence, we have

$$\begin{aligned}
Q_t^j &= \sum_{k\in\mathcal{S}_t^j} \frac{p_t^j(k)}{1+\mu_t^j} + \sum_{k\notin\mathcal{S}_t^j} \frac{p_t^j(k)}{p_t^j(k)+\mu_t^j} \\
&\overset{(a)}{\leq} \sum_{k\in\mathcal{S}_t^j} p_t^j(k)\left(1 - \frac{\mu_t^j}{2}\right) + \sum_{k\notin\mathcal{S}_t^j} \frac{p_t^j(k)}{p_t^j(k)+\mu_t^j} \\
&\leq K - |\mathcal{S}_t^j| + \sum_{k\in\mathcal{S}_t^j} p_t^j(k) - \sum_{k\in\mathcal{S}_t^j} \frac{\mu_t^j p_t^j(k)}{2} \qquad (72)
\end{aligned}$$

where (a) uses the inequality $\frac{1}{1+x} \leq 1 - \frac{x}{2}, \forall x \in [0,1]$. □

F. Proof of Corollary 4

Lemma 5 can be adopted to bound the value of cooperation λ. The upper bound on Q_t^j in Lemma 5 can be rewritten as $Q_t^j \leq \min\{K, K - |\mathcal{S}_t^j| + 1\}$. Plugging the latter into the definition of λ, and using the fact that $\delta \leq K$, we arrive at

$$\lambda \leq \frac{1}{J}\sum_{j=1}^{J}\sqrt{\frac{1}{T} + \frac{1}{KT}\sum_{t=1}^{T}\min\Big\{K, K+1-|\mathcal{S}_t^j|\Big\}}. \quad (73)$$

References

[1] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM Journal on Computing*, vol. 32, no. 1, pp. 48–77, 2002.

[2] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. and Trends® in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[3] N. Cesa-Bianchi, C. Gentile, Y. Mansour, and A. Minora, "Delay and cooperation in nonstochastic bandits," *J. Machine Learning Res.*, vol. 49, pp. 605–622, 2016.

[4] G. Y. Chang, S. Y. Wang, and Y. X. Liu, "A jamming-resistant channel hopping scheme for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6712–6725, Oct. 2017.

[5] L. Chen and J. Xu, "Socially trusted collaborative edge computing in ultra dense networks," in *Proc. of ACM/IEEE Symposium on Edge Computing*, San Jose, CA, Oct. 2017, p. 9.

[6] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Select. Areas Commun.*, vol. PP, no. 99, pp. 1–1, 2018.

[7] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Trans. Signal Processing*, vol. 65, no. 24, pp. 6350–6364, Dec. 2017.

[8] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic IoT management," *IEEE Internet of Things Journal*, to appear, 2018. [Online]. Available: https://arxiv.org/abs/1707.09060

[9] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Networking*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[10] S. M. Cheng, P. Y. Chen, C. C. Lin, and H. C. Hsiao, "Traffic-aware patching for cyber security in mobile IoT," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 29–35, 2017.

[11] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[12] S. D'Oro, E. Ekici, and S. Palazzo, "Optimal power allocation and scheduling under jamming attacks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1310–1323, Jun. 2017.

[13] A. Garnaev, M. Baykal-Gursoy, and H. V. Poor, "A game theoretic analysis of secret and reliable communication with active and passive adversarial modes," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 2155–2163, 2015.

[14] A. Garnaev, Y. Liu, and W. Trappe, "Anti-jamming strategy versus a low-power jamming attack when intelligence of adversary's attack type is unknown," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 1, pp. 49–56, Mar. 2016.

[15] E. Hazan, "Introduction to online convex optimization," *Found. and Trends® in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[16] L. Hu, H. Wen, B. Wu, F. Pan, R. F. Liao, H. Song, J. Tang, and X. Wang, "Cooperative jamming for physical layer security enhancement in internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 219–228, Feb. 2018.

[17] L. Jia, Y. Xu, Y. Sun, S. Feng, L. Yu, and A. Anpalagan, "A game-theoretic learning approach for anti-jamming dynamic spectrum access in dense wireless networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1646–1656, 2018.

[18] P. Joulani, A. György, and C. Szepesvári, "Delay-tolerant online convex optimization: Unified analysis and adaptive-gradient algorithms," in *Proc. of AAAI Conf. on Artificial Intelligence*, vol. 16, Phoenix, Arizona, 2016, pp. 1744–1750.

[19] V. Kanade, H. B. McMahan, and B. Bryan, "Sleeping experts and bandits with stochastic action availability and adversarial rewards," in *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, Clearwater Beach, Florida, April 2009, pp. 272–279.

[20] R. Kleinberg, A. Niculescu-Mizil, and Y. Sharma, "Regret bounds for sleeping experts and bandits," *Machine Learning*, vol. 80, no. 2-3, pp. 245–272, April 2010.

[21] B. Li, T. Chen, and G. B. Giannakis, "Secure edge computing in IoT via online learning," in *Proc. of Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 2018.

[22] ——, "Bandit online learning with unknown delays," in *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, 2019, pp. 993–1002.

[23] B. Li, T. Chen, X. Wang, and G. B. Giannakis, "Real-time energy management in microgrids with reduced battery capacity requirements," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 1928–1938, 2017.

[24] Z. Lu, W. Wang, and C. Wang, "Modeling, evaluation and detection of jamming attacks in time-critical wireless applications," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1746–1759, Aug. 2014.

[25] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 601–628, First Quarter 2018.

[26] T. H. Noor, Q. Z. Sheng, L. Yao, S. Dustdar, and A. H. Ngu, "CloudArmor: Supporting reputation-based trust management for cloud services," *IEEE Trans. Parallel and Distrib. Syst.*, vol. 27, no. 2, pp. 367–380, Feb. 2016.

[27] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, Feb. 2018.

[28] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable IoT architecture based on transparent computing," *IEEE Network*, vol. 31, no. 5, pp. 96–105, 2017.

[29] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, Jan. 2018.

[30] H. B. Salameh, S. Almajali, M. Ayyash, and H. Elgala, "Spectrum assignment in cognitive radio networks for internet-of-things delay-sensitive applications under jamming attacks," *IEEE Internet of Things Journal*, to appear, 2018.

[31] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[32] S. Vakili, K. Liu, and Q. Zhao, "Deterministic sequencing of exploration and exploitation for multi-armed bandit problems," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 759–767, Oct 2013.

[33] S. Vakili and Q. Zhao, "Risk-averse multi-armed bandit problems under mean-variance measure," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 6, pp. 1093–1111, Sept 2016.

[34] B. Wang, Y. Wu, K. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Select. Areas Commun.*, vol. 29, no. 4, pp. 877–889, 2011.

[35] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1784–1797, Mar. 2018.

[36] Q. Wang, P. Xu, K. Ren, and X. Y. Li, "Towards optimal adaptive UFH-based anti-jamming wireless communication," *IEEE J. Select. Areas Commun.*, vol. 30, no. 1, pp. 16–30, Jan. 2012.

[37] X. Xu, S. Vakili, Q. Zhao, and A. Swami, "Online learning with side information," in *Proc. IEEE Military Communications Conference (MILCOM)*, Oct 2017, pp. 303–308.

[38] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[39] L. Zhang, Z. Guan, and T. Melodia, "United against the enemy: Anti-jamming based on cross-layer cooperation in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5733–5747, Aug. 2016.

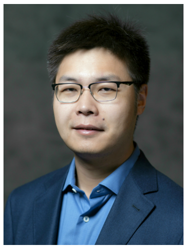
[40] P. Zhou, Q. Wang, W. Wang, Y. Hu, and D. Wu, "Near-optimal and practical jamming-resistant energy-efficient cognitive radio communications," *IEEE Trans. Inf. Forensic Secur.*, vol. 12, no. 11, pp. 2807–2822, Nov. 2017.

[41] Q. Zhu, H. Li, Z. Han, and T. Basar, "A stochastic game model for jamming in multi-channel cognitive radio systems," in *Proc Intl. Conf. on Communications*. IEEE, 2010, pp. 1–6.

[42] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, Sept. 2016.

Bingcong Li received the B. Eng. degree (with highest honors) in Communication Science and Engineering from Fudan University in 2017. He is now pursuing his Ph.D. degree at UMN. He received the National Scholarship twice from China in 2014 and 2015, and UMN ECE Department Fellowship in 2017.

His research interests lie in optimization and reinforcement learning.



Tianyi Chen (S'14) received the B. Eng. degree in Communication Science and Engineering from Fudan University in 2014, the M.Sc. and Ph.D degrees in Electrical and Computer Engineering (ECE) from the University of Minnesota (UMN), in 2016 and 2019, respectively.

Since August 2019, he is with Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute as an Assistant Professor. During 2017-2018, he has been a visiting scholar at Harvard University, University of California, Los Angeles, and University of Illinois Urbana-Champaign.He was a Best Student Paper Award finalist in the 2017 Asilomar Conf. on Signals, Systems, and Computers. He received the National Scholarship from China in 2013, the UMN ECE Department Fellowship in 2014, and the UMN Doctoral Dissertation Fellowship in 2017.

His research interests lie in optimization, machine learning and statistical signal processing with applications to communication, and energy systems.

Georgios B. Giannakis (F'97) received his Diploma in Electrical Engr. from the Ntl. Tech. Univ. of Athens, Greece, 1981. From 1982 to 1986 he was with the Univ. of Southern California (USC), where he received his MSc. in Electrical Engineering, 1983, MSc. in Mathematics, 1986, and Ph.D. in Electrical Engr., 1986. He was with the University of Virginia from 1987 to 1998, and since 1999 he has been a professor with the Univ. of Minnesota, where he holds an Endowed Chair in Wireless Telecommunications, a University of Minnesota McKnight Presidential Chair in ECE, and serves as director of the Digital Technology Center.

His general interests span the areas of communications, networking and statistical signal processing - subjects on which he has published more than 450 journal papers, 750 conference papers, 25 book chapters, two edited books and two research monographs (h-index 143). Current research focuses on learning from Big Data, wireless cognitive radios, and network science with applications to social, brain, and power networks with renewables. He is the (co-) inventor of 30 patents issued, and the (co-) recipient of 9 best paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, the G. W. Taylor Award for Distinguished Research from the University of Minnesota, and the IEEE Fourier Technical Field Award (2015). He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SP Society.