A Game Theoretical Error-Correction Framework for Secure Traffic-Sign Classification

Muhammed O. Sayin, Chung-Wei Lin, Eunsuk Kang, Shinichi Shiraishi, and Tamer Başar, Life Fellow, IEEE

Abstract

We introduce a game theoretical error-correction framework to design classification algorithms that are reliable even in adversarial environments, with a specific focus on traffic-sign classification. Machine learning algorithms possess inherent vulnerabilities against maliciously crafted inputs especially at high dimensional input spaces. We seek to achieve reliable and timely performance in classification by redesigning the input space physically to significantly lower dimensions. Traffic-sign classification is an important use-case enabling the redesign of the inputs since traffic-signs have already been designed for their easy recognition by human drivers. We encode the original input samples to, e.g., strings of bits, through error-correction methods that can provide certain distance guarantees in-between any two different encoded inputs. And we model the interaction between the defense and the adversary as a game. Then, we analyze the underlying game using the concept of hierarchical equilibrium, where the defense strategies are designed by taking into account the best possible attack against them. At large scale, for computational simplicity, we provide an approximate solution, where we transform the problem into an efficient linear program with substantially small size compared to the original size of the entire input space. Finally, we examine the performance of the proposed scheme over different traffic-sign classification scenarios.

Index Terms

Game theory; Autonomous driving; Traffic sign recognition; Adversarial classification; Certifiable machine learning.

I. INTRODUCTION

MACHINE learning is one of the key enabling technologies for autonomous vehicles. An autonomous vehicle can learn how to recognize the surroundings, e.g., traffic-signs, and can base its strategic decisions on this information. Along with the development in such technologies, it is only a matter of time for autonomous systems to achieve remarkable performance, which is comparable with human drivers. However, for the time being, there are still important, yet not completely addressed, challenges for autonomous driving. Traffic-sign classification is one of these challenges. Particularly, varying weather conditions, changing lighting throughout the day, fading colors, and occlusion, e.g., by plants, pose challenges to reliable traffic-sign recognition in real-time [1] while there can also be physical adversarial modifications, e.g., stickers or graffiti, on the traffic-signs [2].

Recently, traffic-sign recognition has attracted substantial attention in the field of intelligent transportation systems [3]–[9]. In [3], the authors have studied convolutional neural networks trained according to hinge loss stochastic gradient descent to achieve fast and stable convergence rates with substantial recognition performance. In [4] and [5], the authors have proposed text-based detection systems for traffic panels that could include information that can vary substantially. Computational complexity of the recognition algorithms plays a significant role for real-time applications since autonomous vehicles are time-critical systems [6], [7]. In [6], the authors have sought to enhance the performance of convolutional neural networks for faster performance in real-time applications through localization of the traffic-signs in the input images based on their types. In [7], the authors have proposed kernel-based extreme learning machines with deep perceptual features to achieve comparable performance to hinge-loss
stochastic gradient based convolutional neural networks (proposed in [3]) with reduced computational complexity. Tree-based hierarchical structures have also been proposed to achieve coarse-to-fine traffic sign detection [9], [9].

Traffic-sign recognition process can be split into traffic-sign detection and traffic-sign classification [1]. The former focuses on locating the traffic signs at a given image frame while the latter seeks to categorize the located traffic sign correctly. Different from the previous works [3]–[9], in this paper, we seek to address traffic-sign classification in adversarial environments, e.g., when there is a strategic attacker who modifies the traffic sign physically via stickers or graffiti as in [2]. Particularly, a fundamental assumption in machine learning algorithms is that training and testing data are drawn from the same distribution. However, this assumption does not hold in adversarial environments where the testing data could have been crafted strategically. Correspondingly, the performance of machine learning algorithms could degrade significantly in adversarial environments [10]. Reliable performance of machine learning algorithms, even in adversarial environments, is essential for their use in safety-critical systems, e.g., autonomous vehicles. Therefore, the important question that arises is whether an attacker could craft the input through perturbations that are imperceptible for humans (i.e., a human would still easily classify the input correctly) while the algorithm classifies the input as the attacker’s targeted class. An adversarial example is an input sample generated by perturbing a correctly classified input with imperceptible change to lead to misclassification [10]. In the following subsection, we briefly review the extensive literature on adversarial examples for machine learning algorithms before motivating our own game theoretical approach to the issue.

A. Adversarial Classification

Robustness of machine learning algorithms in adversarial environments has been studied extensively over the last decade, e.g., [2], [10]–[13]. However, it still has remained an open problem in spite of the significant progress in understanding the intriguing properties of these algorithms. Interested readers can refer to the key references [10]–[12]. At high-dimensions, adversarial examples exist virtually near all the inputs although they are rare with respect to the underlying distribution [10]. It has been shown that the issue is neither over-fitting on the training data, under-training due to the limited number of training samples nor the nonlinearity inherent to machine learning algorithms [11]. Indeed, in [11], the authors advocate that even linear behavior, e.g., additive iterations, in high-dimensional spaces is sufficient for vulnerability against maliciously crafted inputs while the linear nature plays a significant role in the computational simplicity of learning algorithms. Correspondingly, the authors draw the following conclusion: all machine learning algorithms, from simple linear classification to complex deep neural networks, are vulnerable to adversarial examples at high dimensions [11].

Recently, substantial amount of defense methods have been proposed to make machine learning algorithms robust against adversarial scenarios. These defense methods have been developed to provide robustness against a certain class of attacks, and it is possible to bypass them easily by small modification of the attacks [12]. An effective defense strategy is one that is obtained by introducing adversarial examples into the training data, however, this requires the computation of the adversarial examples while the adversarial scenario generation problem is highly non-linear and non-convex [14]. For scalability, gradient-based attacks can be used to train the algorithms. However, such defense strategies face the obfuscated gradient phenomenon, in which they can be bypassed through the slight modification of the attack away from the gradient [12]. On the other hand, certification based approaches seek to guarantee that the input would be classified correctly as long as a certain norm of the perturbation is bounded from above by a certain constant [15]. Such approaches are not scalable for high-dimensional input spaces and the choice of the norm for imperceptible perturbations is a subjective matter. As an example, there can be adversarial examples that perturb the image substantially while preserving its semantic such that a human could still easily classify it correctly [16].

Furthermore, adversarial scenarios are not limited to the digital world. In the physical-world, e.g., traffic-sign recognition, which is of primary concern in this paper, objects can also be crafted physically in order to lead to misclassification in vision-based systems. In [17], the authors have claimed that such adversarial modifications in real physical implementations are ineffective due to the uncertainty on the distance or the angle of the view. However, in [2], [13], the authors have shown that the physical-attack can be launched in a robust way against such unknown transformations. To this end, they have modeled such unknown transformations synthetically (and also experimentally in [2]) as a distribution of transformations, and computed robust adversarial examples that perform successfully in expectation over these transformations. In this paper, we will seek to detect such robust physical attacks on traffic signs.
B. Game Theoretical Approaches

Substantial amount of defense methods have already been proposed to make machine learning algorithms robust against attacks. However, they all could easily be bypassed by small modification of the attacks [12]. Therefore, a game theoretical equilibrium analysis is essential for reliable classification, since at an equilibrium, the defender would be defending against the strongest (worst) possible attack for the proposed defense. Particularly, at a Nash equilibrium, the players, i.e., the defender or the attacker, would be playing in the best way against each other, and neither player would accrue benefit by deviating from its action unilaterally [18], [19]. Therefore, we seek to design secure traffic-sign classification algorithms against attacks that are robust against unknown transformations in physical applications by taking into account also the adaptation of the attacks to any change in the defense strategies, i.e., within a game theoretical framework.

This paper is definitely not the first one approaching the adversarial classification problem through a game-theoretical lens. For example, in [20], the authors have introduced a non-zero sum game between an attacker and a classifier; however, they have not studied any notion of equilibrium. In [21], [22], the authors have studied adversarial prediction problems for a certain class of learners, e.g., support-vector-machines, in terms of Nash and Stackelberg equilibria, respectively.

Recently, in [23], the authors have analyzed adversarial binary-classification as a non-zero sum game between an attacker and a classifier. The classifier seeks to detect whether the input is coming from the attacker or from a known benign-distribution. On the other side, the attacker seeks to maximize his reward (which depends on the input) without being detected by the classifier. The authors have shown that the classifier can restrict the strategies to mixtures of classifiers setting threshold on the reward of the attacker. However, the results in [23] cannot be extended to reliable classification under physical attacks since physical attacks are subject to random transformations, e.g., due to the distance or the angle of view [2], [13], [17]. Due to that randomness, different attacks with different rewards can lead to the same input. Therefore, given the input, the defender cannot know the attacker’s reward to compare it against a threshold.

C. Motivation

We emphasize that the main issue is that an adversary can cause misclassification through small perturbations on inputs in image classification, in general. Clearly, an adversary can always cause misclassification by modifying the input substantially, however, that would be costly for the adversary. On the other hand, small perturbations would have relatively negligible cost compared to substantial modifications. Therefore, the ability to cause misclassification with relatively negligible cost could incentivize such attacks. We can therefore deter such attacks by making the misclassification costly for the attacker.

Furthermore in [10], the authors have advocated that machine learning algorithms are that much vulnerable to attacks because of the high-dimensional input space and linear nature of the algorithms for computational simplicity. Here, we target the former one by reducing dimension of the input space for reliable traffic-sign classification. Targeting the latter one by increasing the complexity of the algorithms might not be desirable for implementation in time-critical systems, such as autonomous vehicles. Note also that traffic signs are designed to facilitate their classification by human drivers [24]. However, not only human drivers will be in the traffic in the near future, but also autonomous vehicles will be gradually introduced into it. In that respect, the traffic signs could be designed to facilitate their classification also by autonomous vehicles. We emphasize that we do not have the opportunity to design the input space in general classification problems. Traffic-sign classification is an important use-case. We can include bar code like identifiers on the traffic-signs for their reliable classification (which is analytically guaranteed) by autonomous vehicles. However, in pedestrian detection, we may not, for example, ask pedestrians to wear certain type of clothing in general.

In order to reduce the dimension of the input space, we can view the traffic-sign classification problem from a wider perspective as a communication problem. The traffic sign and human driver/autonomous vehicle can be viewed as a transmitter and a receiver, respectively. Then, the message, i.e., the type of the traffic sign, would be transmitted over a noisy channel that can lead to errors in the transmitted message. We note that in coding theory, redundancy is an important tool for error correction [25]. As an example, maximum distance separable codes, e.g., Reed-Solomon code [26], can maximize the guaranteed lower bound on the perturbations that can cause any misclassification by designing the redundancy introduced intelligently. Correspondingly, we can benefit
from error-correction methods in traffic sign recognition for security when we encode the traffic-signs into, e.g., barcode like strings of bits. Furthermore, a game-theoretical model can lead to guarantees beyond error-correction since the errors could be designed strategically by an adversary rather than being generated randomly by nature.

We also emphasize that even though the information about the traffic signs could also be included in the navigation maps, the global positioning system (GPS) may not be able to provide reliable information in certain urban or rural areas, e.g., Manhattan area in New York City [27]. The information provided by traffic signs could also be dynamic, and any change, e.g., due to road construction, can lead to serious synchronization issues.

D. Our Contributions

In this paper, we propose reliable classification algorithms for localized traffic signs with theoretically guaranteed performance against physical attacks. To this end, we exploit the ability to design the traffic signs physically so that we can transform the problem into the framework of error-correction methods that can provide certain formal guarantees over the perturbations that could cause misclassification. With a linear error-correcting block code, we encode each traffic sign into codewords, which can be patched on traffic signs as exemplified in Fig. 1. Error-correction methods are powerful against random perturbations. For effective performance against advanced (strategic) attackers and in order to reduce the feasibility of such attacks, we design a detection strategy within a game theoretical framework, where the attacker would attack the best possible way against the proposed defense.

Our goal is to increase the cost to the attacker at the expense of false alarms. The attacker perturbs the codewords, e.g., via stickers, at the expense of attack complexity, in order to cause misclassification evasively. We note that the codewords perturbed by the attacker could also be perturbed by random noise as in [2] and correspondingly, an advanced attacker attacks in a robust way by taking into consideration such noisy perturbations on the crafted codeword. On the other side, based on the distance between the received codeword and the closest encoded codeword, i.e., corresponding to a traffic sign, our goal, as a defense mechanism, is to detect the attacker. The defense policy is based on only that distance instead of the received codeword in order to avoid computational issues due to large dimensionality. We design a randomized detection algorithm under the conservative assumption that the attacker is aware of the designed detection probabilities and attacks in the best possible way.

We specifically consider a nonzero-sum game setting, rather than a zero-sum one, between the defense and the adversary since the defense seeks to maximize the attacker’s cost at the expense of false alarms. Correspondingly, we consider a hierarchical setting and analyze the Stackelberg equilibrium [19], where the equilibrium would have unique value if it exists, rather than Nash equilibrium where multiple equilibria may or may not exist. We observe that this problem is strategically equivalent to a zero-sum game. Then, using this property, we show the existence of a Stackelberg equilibrium. Even though we have considered a defense policy based on that distance, the attack space is very large. And due to the hierarchical setting, the defense needs to anticipate the attacker’s best reaction to any proposed strategy. To this end, we further examine the attacker actions and show that the attacker can be viewed as selecting an action from a quotient space of the actual action space with respect to a certain equivalence relation in Subsection IV-A. However, that quotient space can still be large to search over with many traffic signs. Accordingly, we provide a method to relax the attack space to address such computational issues in Subsection IV-C. This relaxation enables us to transform the problem into an efficient linear program (LP) with substantially smaller size in Subsection IV-D. Finally, we also analyze the performance numerically over various scenarios.

Our main contributions are as follows:

Fig. 1: Example implementations on the stop sign, which is only limited by imagination and creativity of the designer.
Fig. 2: A game between attacker ($A$), and defender ($D$). For each traffic sign, there exists a unique codeword. $A$ selects which traffic sign to attack and attacks by changing the symbols in the codeword physically. $D$ observes a noisy version of the codeword which might have been perturbed by $A$ and seeks to detect whether there has been an attack or not.

- To the best of our knowledge, this is the first work in the literature to incorporate a game theoretical error-correction code framework to the classification problem. This leads to formal guarantees against the required perturbation and the attackers’ adaptation to the proposed defense.
- We provide an LP-based algorithm to compute the optimal (scalable) randomized detection strategy, which can reduce the verification complexity.
- We examine the performance of the proposed scheme over different traffic-sign classification scenarios.

The paper is organized as follows: In Section II we provide preliminary information about error-correction coding. In Section III we formulate the problem for secure traffic-sign classification against physical attacks. In Section IV we analyze the equilibrium for the robust adversarial classification game. We examine the performance of the proposed framework over different traffic-sign classification scenarios in Section V. We conclude the paper in Section VI with several remarks and identifying possible research directions.

**Notation:** We denote sets by calligraphic letters, e.g., $S$. For a set $S$, $|S|$ denotes its cardinality. For the sets $A$ and $B$, $A^B$ denotes the power set. For logical propositions, $\lor$ and $\land$ denote disjunction and conjunction operations, respectively. $\Delta^d$ denotes the $d$-simplex.

**NOMENCLATURE**

**Problem Setting:**

- $[n, k, d]_q$ linear block code
- $n \in \mathbb{Z}$ codeword length
- $k \in \mathbb{Z}$ message length
- $d \in \mathbb{Z}$ (minimum) distance of the code
- $d_o = \lfloor (d - 1)/2 \rfloor$ error-correction diameter
- $\Sigma$ alphabet of the symbols
- $q = |\Sigma|$ alphabet size
- $\mathcal{T}$ enumeration of all traffic signs
- $\mathcal{T}^e = \mathcal{T} \cup \{-1\}$ extended set of traffic signs
- $\Sigma^n$ set of all codewords
- $\Sigma^n_o \subset \Sigma^n$ set of encoded codewords
- $g : \mathcal{T} \rightarrow \Sigma^n_o$ encoding mapping
- $h : \Sigma^n \rightarrow \mathcal{T}^e$ decoding mapping
- $t \in \mathcal{T}$ a traffic-sign
- $\hat{t} \in \mathcal{T}^e$ decoder output
- $x_o \in \Sigma^n_o$ encoded codeword
- $y \in \Sigma^n$ received (noisy) codeword
Error-correcting codes provide certain formal guarantees for the transmission of digital data over noisy channels. We specifically consider algebraic block codes, e.g., Reed-Solomon codes \cite{26}, which encode data in blocks. Formally, a block code operates over a finite alphabet of symbols, denoted by \(\Sigma\), and maps \(k \in \mathbb{Z}\) symbols, i.e., a message \(m \in \Sigma^k\), to \(n \in \mathbb{Z}\) symbols, i.e., a codeword \(c \in \Sigma^n\). The (minimum) distance of a block, denoted by \(d \in \mathbb{Z}\), is the minimum number of positions where any two distinct codewords \(c_1 \neq c_2\) differ, i.e., the Hamming distance \cite{25} in-between the distinct codewords over the \(n\)-dimensional space \(\Sigma^n\). Note that the minimum distance \(d\) implies that the block code can detect \(d - 1\) symbol errors and correct up to \(d_0 := \left\lfloor \frac{d - 1}{2} \right\rfloor\) symbol errors since there exists no other codeword within \(d - 1\) diameter of each codeword over \(\Sigma^n\).
by \([n, k, d]_q\). The Reed-Solomon code is a \([n, k, n-k+1]_q\) code\(^1\) and is a maximum distance separable code, which maximizes the minimum distance between any two distinct codewords within the general class of linear codes with codeword length \(n\) and message length \(k\). Note that Reed-Solomon codes provide effective guarantees against symbol errors and correspondingly effective against contiguous bit errors, e.g., can be effective against obfuscation by plants or graffiti or adversarial stickers on traffic-signs [2]. As an example, QR codes constitute a widely-used application of Reed-Solomon codes.

Consider that the number of symbol errors, denoted by \(e \in \mathbb{Z}\), is more than half of the minimum distance, i.e., \(e > d_o\). We say that a decoder error exists if the Hamming distance between the received corrupted codeword and any other codeword is less than or equal to \(d_o\). Further, we say that a decoder failure exists if the Hamming distance between the received corrupted codeword and all the other codewords is more than \(d_o\) [28].

### III. Problem Formulation

Consider two players: an attacker (A) and a defender (D), as seen in Fig. 2 playing a secure traffic-sign classification game (which will be defined in detail later in this section), where D seeks to detect any intervention by A while A seeks to perturb the traffic signs to lead to misclassification without being detected. Consider a linear block code \([n, k, d]_q\) with length \(n \in \mathbb{Z}\), message length \(k \in \mathbb{Z}\), minimum distance \(d \in \mathbb{Z}\), alphabet \(\Sigma\), and alphabet size \(q = |\Sigma|\). Let \(T\) denote the enumeration of all traffic signs. Traffic signs are encoded into codewords \(x_o \in \Sigma \to \Sigma^n\) via the corresponding encoding mapping

\[
g : T \to \Sigma^n.
\]

The observed codeword by the decoder can be a noisy version of the true codeword due to random perturbations by nature, e.g.,

- varying weather conditions,
- changing lighting conditions throughout the day,
- fading colors,
- occlusion, e.g., by plants or graffiti,
- uncertainty in the distance or angle of the view.

Analogously, this can be viewed as arising when the codewords are transmitted to the decoder over a noisy channel with a probability transition mapping \(p(y|x)\) corresponding to the probability of receiving codeword \(y \in \Sigma^n\) given that the transmitted codeword is \(x \in \Sigma^n \to \Sigma^n\). We consider that all the symbol errors by nature are equally likely and independent of each other. Let \(p_e \in [0, 1]\) denote the probability that there can be an error in a symbol. Suppose also that in a symbol error, the change of the symbol to any other symbol in the alphabet is equally likely. Recall that a codeword \(x \in \Sigma^n\) consists of \(n\) symbols. A received codeword \(y \in \Sigma^n\) is, then, decoded by a certain decoding mapping

\[
h : \Sigma^n \to T^e,
\]

where \(T^e := T \cup \{-1\}\) and the output \(-1\) implies that the received codeword is not decodable, i.e., that codeword is not within \(d_o\) diameter of any codeword in \(\Sigma^n\).

A can select which traffic sign to attack. Let \(a_t \in T\) denote the attacked sign. Then, A can craft \(g(a_t) \in \Sigma^n\) to \(a_x \in \Sigma^n\) by introducing error in order to control the decoder output. The cost of crafting \(g(a_t) \in \Sigma^n\) to \(a_x \in \Sigma^n\) is the Hamming distance between the codewords \(g(a_t), a_x \in \Sigma^n\), and denoted by \(\eta : T \times \Sigma^n \to \mathbb{Z}\), i.e., we have

\[
\eta(a_t, a_x) = H(g(a_t), a_x).
\]

We denote A’s action space by \(A := T \times \Sigma^n\) and denote A’s action by \(a := (a_t, a_x)\). A can select a mixed strategy \(\alpha = \{\alpha_a\} \in \Delta^{|A|}\) over A such that \(\alpha_a\) denotes the probability of taking action \(a = (a_t, a_x) \in A\), i.e., attacking the sign \(a_t \in T\) by crafting the associated codeword to \(a_x \in \Sigma^n\).

If the received codeword is decodable, based on the number of symbol errors, D can report an issue against the possibility of adversarial intrusion so that further (costly) investigations can take place. We consider the scenario where further investigations always take place for not decodable codewords. D designs a randomized detection rule

\[
\pi : \{1, \ldots, d_o\} \to [0, 1],
\]

\(^1\)The alphabet size is in general a prime power and length of codeword \(n < q\), e.g., often \(n = q - 1\).
where if $y \in \Sigma^n$ is decodable, letting
\[ \hat{y} := g(h(y)) \]
be the codeword in $\Sigma^n_0$ closest to the decodable $y$, then $\pi(H(y, \hat{y}))$ corresponds to the probability of triggering an alarm for the given noisy codeword $y \in \Sigma^n$. For notational simplicity, we introduce
\[ \tilde{\pi}(y) := \begin{cases} 1 & \text{if } h(y) = -1, \\ \pi(H(y, \hat{y})) & \text{else} \end{cases} \]
Furthermore, there exists a loss function $\ell : \mathcal{T} \times \mathcal{T}^e \to \{0, 1\}$, where, e.g.,
\[ \ell(t, \hat{t}) = \mathbb{1}_\{t = \hat{t}\}, \]
corresponds to the loss for decoding the true traffic sign $t \in \mathcal{T}$ as $\hat{t} \in \mathcal{T}^e$, which can be viewed as the cost of not being able to classify the input accurately.

**Remark (Scalable Defense).** We consider a randomized detection rule depending on the number of symbol errors for scalability. In particular, for such a detection rule, $D$ selects a vector over the space $[0, 1]^{d_o}$ while anticipating $A$’s possible attack against the selected detection rule. On the other hand, if $D$ were to select a (randomized) detection rule based on the received codeword, then $D$ would select a vector over the space $[0, 1]^{2^n}$, which is $q^n$ dimensional and exponential in the number of symbols, i.e., $n$, whereas $d_o \ll q^n$ is linear in $n$.

Both players know (or have learned) the underlying distributions, as well as the encoding and decoding mappings. $A$ seeks to maximize the expected loss while evasively seeking to minimize the expected detection cost at the expense of attack complexity. Therefore, $A$ selects the mixed strategy $\alpha \in \Delta^{|A|}$ in order to minimize the cost function:
\[ U_A(\alpha, \pi) := \sum_{a \in A} \alpha_a \eta(a, a_x) - \gamma^F \sum_{a \in A} \sum_{y \in \Sigma^n} \ell(a, h(y)) p(y|a_x) \alpha_a + \gamma^D \sum_{a \in A} \sum_{y \in \Sigma^n} \tilde{\pi}(y) p(y|a_x) \alpha_a, \]
where $\gamma^D, \gamma^F > 0$ are certain multiplication factors corresponding, respectively, to the cost due to detection and the gain due to the misclassification. Detection is more costly for $A$ than the gain due to misclassification. Note that minimization of the expected cost due to the uncertainty of the channel is in-line with the expectation-over-transformation framework proposed in [13] as a response to the claimed ineffectiveness of attacks in real implementations [17]. The attackers can generate robust attacks by considering the expected impact of the uncertainties due to the channel [13].

On the other side, $D$ seeks to detect the adversarial intrusion by selecting a detection rule that maximizes $A$’s cost at the expense of false alarms. $D$ takes action $\pi \in [0, 1]^{d_o}$ in order to minimize the cost function:
\[ U_D(\alpha, \pi) := -U_A(\alpha, \pi) + \gamma^F \sum_{t \in \mathcal{T}} \sum_{y \in \Sigma^n} \tilde{\pi}(y) p(y|g(t)) p(t), \]
where $\gamma^F > 0$ is certain multiplication factor corresponding to the cost of false alarm.

Furthermore, we consider a hierarchical setting, where $A$ can know (or learn) $D$’s randomized detection algorithm, in order to avoid obscurity based defense, which can be bypassed when an advanced attacker learns the information in obscurity. Therefore, this interaction can be modeled as a Stackelberg game [19], where $D$ is the leader. Therefore, the **secure traffic-sign classification game**
\[ \mathcal{G} := (\Delta^{|A|}, [0, 1]^{d_o}, \ell(\cdot), \eta(\cdot), p(y|x), p(t), \gamma^F, \gamma^D, \gamma) \]
is a Stackelberg game between $A$ and $D$, where $D$ is the leader and $A$ is the follower. Their (finite) action spaces are given by $\Delta^{|A|}$ and $[0, 1]^{d_o}$, respectively. The players seek to minimize their cost functions (9) and (10), respectively. Since $A$ is the follower and takes actions knowing $D$’s action $\pi \in [0, 1]^{d_o}$, we let $B(\pi) \in \Delta^{|A|}$ be $A$’s reaction set to $D$’s action $\pi \in [0, 1]^{d_o}$. Then, the action and best reaction pair $(\pi^*, B(\pi^*))$ attains the Stackelberg equilibrium provided that
\[ \pi^* = \arg\min_{\pi \in [0, 1]^{d_o}} \max_{\alpha \in B(\pi)} U_D(\alpha, \pi) \]
\[ B(\pi) = \arg\min_{\alpha \in \Delta^{|A|}} U_A(\alpha, \pi). \]

In the following section, we analyze the equilibrium to the secure traffic-sign classification game $\mathcal{G}$. 
IV. SECURE TRAFFIC-SIGN CLASSIFICATION GAME

In order to compute the action and the best reaction pair \((\pi^*, B(\pi^*))\) attaining the Stackelberg equilibrium (11), we first show the existence of an equilibrium by showing that the game \(G\) is strategically equivalent to a zero-sum game. \(A\)'s action space is relatively much larger than \(D\)'s action space. Since \(D\) needs to anticipate \(A\)'s best reaction to any selected detection rule, our goal is to examine \(A\)'s best response in order to formulate certain equivalence classes, where all of \(A\)'s actions in a class lead to the same game outcome (see, Subsection IV-A). However, depending on the size of the input space, i.e., \(\Sigma^n\), we may need to further examine \(A\)'s best reaction in order to avoid computational issues due to large dimensions. We provide an approximation on \(A\)'s best reaction by relaxing the constraints on \(A\)'s action space, which will lead to much more powerful attacker than in practice (see, Subsection IV-C). This yields a conservative defense, which leads to lower cost against the actual attacker with less power in run-time applications. Finally, we transform the problem into an efficient LP with substantially smaller size compared to \(\Sigma^n\) rather routinely (see, Subsection IV-D). We now provide the details of each of these steps.

We first note that without any loss of generality, \(A\)'s objective function can be written as

\[
U_A(\alpha, \pi) - \gamma D \sum_{t \in T} \sum_{y \in \Sigma^n} \pi(y)p(y|g(t))p(t),
\]

instead of (9), since the second term in (12) does not depend on \(A\)'s action \(\alpha \in \Delta^{|A|}\). Then, \(A\)'s objective could be viewed as the complete opposite of \(D\)'s objective (10). Therefore, the game \(G\) is strategically equivalent to a zero-sum game between \(A\) and \(D\). And correspondingly, the best action for \(D\) is given by

\[
\pi^* = \arg\min_{\pi \in [0,1]^d} \max_{\alpha \in \Delta^{|A|}} U_D(\alpha, \pi),
\]

since for a given \(\pi \in [0,1]^d\), all the actions in the reaction set of \(A\), i.e., \(B(\pi)\), lead to the same cost for \(D\) due to the zero-sum structure. Based on this, the following proposition shows that there exists an equilibrium to the game \(G\).

**Proposition 1** (Existence Result). There exists a pair of \(D\)'s (pure) action and \(A\)'s reaction \((\pi^*, B(\pi^*))\) attaining the Stackelberg equilibrium \(G\), i.e., satisfying (11).

**Proof.** The observation that the best \(D\) action can be computed by (13) yields that if we can show the existence of a solution for (13), then there exists an equilibrium to the game \(G\). However, existence of a solution for (13) is not guaranteed without certain continuity conditions on \(U_D(\alpha, \pi)\) since the constraint sets \([0,1]^d\) and \(\Delta^{|A|}\) are not finite sets. Even though the complexity \(\eta(\cdot)\) and loss \(\ell(\cdot)\) are not continuous functions of their arguments, the optimization objective \(U_A(\alpha, \pi)\) is linear, and correspondingly, continuous in the optimization arguments \(\alpha \in \Delta^{|A|}\) and \(\pi \in [0,1]^d\). Since the objective function is continuous in the optimization arguments and the constraint sets are decoupled, the maximum theorem [29] yields that

\[
\max_{\alpha \in \Delta^{|A|}} U_D(\alpha, \pi)
\]

is a continuous function of \(\pi \in [0,1]^d\). Then, since \([0,1]^d\) is a compact set, the extreme value theorem yields that there exists a solution for (13).

Next, our goal is to compute the best detection rule \(\pi^* \in [0,1]^d\). To this end, \(D\) needs to anticipate \(A\)'s reaction to any selected detection rule, i.e., (14). However, \(A\) selects a mixed strategy \(\alpha \in \Delta^{|A|}\) and the simplex \(\Delta^{|A|}\) has dimension \(q^{n+k} - 1\), which is exponential in \(n + k\) and finding the best reaction, i.e., a vector in that space, is computationally demanding for long codewords and messages even for linear algorithms. In the following, we show that we can reduce \(A\)'s action space without any loss of generality.

A. Equivalence Classes on \(A\)'s Best Response

As illustrated in Fig. 3 figuratively, we can categorize \(A\)'s actions into four main classes of strategies:

- Strategy-0 corresponds to the scenario where \(A\) does not attack, i.e., does not introduce any error to any codeword in \(\Sigma_0^n\).
- Strategy-1 corresponds to the scenario where \(A\) introduces symbol error(s) to the codeword of the selected traffic sign, however, the corrupted codeword is still in the decodable region of the associated codeword. Therefore, without
Strategy-0

Fig. 3: Figurative illustration of \( \Sigma^n \) for the Reed-Solomon Code \([7,3,5]_q\). Suppose the attacked codeword is \( g(a_t) = x^1_o \). Decodable regions for the codewords \( x^1_o, x^2_o, x^3_o \in \Sigma^n_o \) are shaded and arcs correspond to the levels of symbol errors. The color coded arrows illustrate figuratively how the corresponding level of symbol error would change with additional nature-induced noisy perturbation of the crafted codeword.

Further corruption by nature-induced random noise, the codeword would be classified accurately. Depending on the random noise, with certain probability, the noisy codeword could enter the decodable region of another codeword, which would lead to misclassification loss and be desirable for \( A \), or could become not decodable or could continue to stay in the decodable region of the associated codeword. The last two possibilities are not desirable for \( A \). However, \( A \) could also be detected within the decodable region of the other codeword depending on \( D \)'s detection rule. We note that cancellation of \( A \)-induced error(s) by nature is less likely yet not impossible.

- Strategy-2 corresponds to the scenario where \( A \) introduces symbol errors to the codeword of the selected traffic-sign and the corrupted codeword becomes not decodable. Without further corruption by nature-induced random noise, the attack would be detected. Similarly, depending on the noise, with certain probability, \( A \) could, or not, lead to a misclassification loss, or could also be detected within the decodable region of a codeword depending on \( D \)'s detection rule.

- Finally, Strategy-3 corresponds to the scenario where \( A \) introduces (relatively more) symbol errors to the codeword of the selected traffic sign and the corrupted codeword moves to the decodable region of another codeword. Depending on the random noise, with certain probability, the noisy codeword could enter the decodable region of another codeword (with smaller probability to return back to the original codeword) or could become not decodable or could continue to stay in the same decodable region. Note that staying at the same decodable region or moving to another one, except to the original one, leads to the same misclassification loss, and therefore is equally preferable by \( A \). However, targeting the misclassification through certain codewords could be more preferable depending on the randomized detection rule.

Based on these strategies, we observe that if there were no noise on \( A \)-corrupted codewords, then Strategy-0, i.e., not attacking, would strictly dominate Strategies 1 and 2 since \( \gamma^D_A > \gamma^L_A \). Furthermore, \( A \) would only play Strategy-3 by deciding the number of symbol errors within the decodable region of the codeword targeted for misclassification since \( D \)'s detection rule depends on the Hamming distance. Furthermore, for a given codeword targeted for misclassification, \( a_t \in T \) could have been selected such that the Hamming distance between \( g(a_t) \) and \( g(h(a_x)) \) is the smallest so that the attack complexity is the lowest. Suppose we have eliminated the actions dominated by Strategy-0 in \( A \). Then, \( A \)'s action space can be written as

\[
A^d_o := \{ (a_t, a_x) \in A | a_x = g(a_t) \lor (h(a_x) \neq -1 \land h(a_x) \neq a_t) \land a_t = \min_{t \in T} H(g(t), h(a_x)) \}. 
\]
Therefore, \(\mathcal{A}\) can mix over the quotient set \(A_o^d/\sim_o\) with respect to the following equivalence relation:

\[
(a_t, a_x) \sim_o (a_t', a_x') \iff \left\{ \begin{array}{l}
H(a_x, g(h(a_x))) = H(a_x', g(h(a_x'))) \\
\land h(a_x) = h(a_x')
\end{array} \right.
\]

The next question we seek to address is “whether we have a similar case for the scenarios where \(\mathcal{A}\)’s crafted codewords could be perturbed by random noise”. In Fig. we observe that Scenario-0 does not necessarily dominate Scenarios 1 and 2 if \(\mathcal{A}\)-corrupted codeword could also be perturbed by random noise. Note that the probability transition mapping can be written as

\[
p(y|x) = \left( \frac{p_r}{q-1} \right)^{H(y,x)} (1-p_e)^n-H(y,x),
\]

which only depends on the distance in-between \(y \in \Sigma^n\) and \(x \in \Sigma^n\). Therefore, there exists

\[
\rho : \{0,1,\ldots,n\} \times \{0,1,\ldots,d_o\} \rightarrow [0,1]
\]

corresponding to the probability that a codeword moving to a certain symbol-error level of another codeword due to random noise since only the distance in-between matters. \(\rho(\cdot)\) can be computed based on combinatorics analytically or using the Monte Carlo method numerically. In terms of \(\rho(\cdot)\), we obtain

\[
\sum_{y \in \Sigma^n} \ell(g(a_t), h(y))p(y|a_x) = \sum_{y \in \Sigma^n} 1_H(y,g(a_t))p(y|a_x)
\]

\[
= 1 - \sum_{k=0}^{d_o} \rho(H(g(a_t), a_x), k).
\]

Furthermore, we have

\[
\sum_{y \in \Sigma^n} \tilde{\pi}(y)p(y|x) = \sum_{x_o \in \Sigma^n} \sum_{k=1}^{d_o} \pi(k)\rho(H(x, x_o), k) + \left( 1 - \sum_{x_o \in \Sigma^n} \sum_{k=0}^{d_o} \rho(H(x, x_o), k) \right),
\]

where the second term (in parantheses) on the right-hand-side is the probability that the corrupted codewords are not in the decodable region of any encoded codeword. We can also write (18) as

\[
\sum_{y \in \Sigma^n} \tilde{\pi}(y)p(y|x) = 1 - \sum_{x_o \in \Sigma^n} \sum_{k=1}^{d_o} (1 - \pi(k))\rho(H(x, x_o), k) - \sum_{x_o \in \Sigma^n} \rho(H(x, x_o), 0).
\]

Therefore, the cost functions depend on \(a_x \in \Sigma^n\) in terms of the distance between \(a_x \in \Sigma^n\) and all the other encoded codewords, i.e., \(x_o \in \Sigma^n\). This differs from the scenario where \(\mathcal{A}\)-corrupted codewords are not perturbed by random noise since with noisy perturbations, not only the distance to the closest encoded codeword but also all the distances to other encoded codewords have an impact on the cost function.

We also observe that \(\mathcal{A}\) can select \(a_t \in T\) irrespective of \(D\)’s detection rule. Particularly, for given distances of \(a_x \in \Sigma^n\) to all encoded codewords, i.e.,

\[
H_o(a_x) := \{ H(x_o, a_x) \}_{x_o \in \Sigma^n},
\]

the optimal \(a_t \in T\) is given by

\[
a_t^* \in \arg\min_{a_t \in T} \eta(a_t, a_x) - \gamma_o \sum_{y \in \Sigma^n} \ell(a_t, h(y))p(y|a_x)
\]

\[
= \arg\min_{a_t \in T} H(g(a_t), a_x) - \gamma_o \left( 1 - \sum_{k=0}^{d_o} \rho(H(g(a_t), a_x), k) \right).
\]

Therefore, by (17), (18), and (21), the cost function \(U_o(\cdot)\) depends on \(a_x \in \Sigma^n\) in terms of the distances between \(a_x \in \Sigma^n\) and the encoded codewords \(x_o \in \Sigma^n\) while the specific identity of the encoded codewords does not impact the cost function. Correspondingly, any permutation of the distances across the encoded codewords would lead to the same cost.

The following lemma recap these results to formulate the equivalence classes on \(\mathcal{A}\)’s best response.
Lemma 1. Consider the action space
\[ \mathcal{A}^d = \{(a_t, a_x) \in \mathcal{A} | a_t \text{ is given by (21)} \}. \] (22)

Then, without loss of generality, instead of mixing over \( \mathcal{A} \), \( \Lambda \) can mix over the quotient set \( \mathcal{A}^d/\sim \) with respect to the following equivalence relation:
\[ (a_t, a_x) \sim (a_t', a_x') \iff \exists \sigma: \{0, \ldots, n\} \to \{0, \ldots, n\} \\land H_o(a_x) = \sigma(H_o(a_x')) \], (23)

where \( \sigma(\cdot) \) denotes a permutation mapping and \( H_o: \Sigma^n \to \{0, 1, \ldots, n\}^{\Sigma_n} \) denotes a certain ordering of the distances to all the encoded codewords, defined in (20).

B. Equilibrium in Compact Form

Note that by (21), there exists a certain \( a_t \in \mathcal{T} \) for given \( a_x \in \Sigma^n \). Therefore, we introduce
\[ r(H_o(a_x)) := \min_{x, o \Sigma^n_0} H(x_o, a_x) + \gamma^L \sum_{k=0}^{d_o} \rho(H(x_o, a_x), k) - \gamma^L, \]

which can be viewed as the reward of \( \Lambda \) when \( a_x \in \Sigma^n \) is played. Then, after some algebra, we can write \( D \)'s cost function in a compact form as
\[ U_D(\alpha, \pi) = \gamma^D_D 1^T \Phi \alpha - r' \alpha - \gamma^D_D \left[ \begin{array}{c} 0' \\ \pi \end{array} \right] \Phi \alpha + \gamma^F_D \left[ \begin{array}{c} 0' \\ \pi \end{array} \right] \Phi_o p_o + \xi, \]

where
\[ \xi := \gamma^F_D - \gamma^D_D - \gamma^F_D 1^T \Phi_o p_o, \]

for certain orderings over the finite sets \( \mathcal{A}^d/\sim \) and \( \Sigma^n_0 \), we define \( \kappa := |\mathcal{A}^d/\sim|, \tau := |\Sigma^n_0| \), column vectors:
\[ \mathbf{r} := \left[ r(H_o(a^1_x)) \ldots r(H_o(a^n_x)) \right]', \mathbf{\alpha} := \left[ \alpha_1 \ldots \alpha_\kappa \right]', \mathbf{\pi} := \left[ \pi(1) \ldots \pi(d_o) \right]', \mathbf{p}_o := \left[ p(x^1_o) \ldots p(x^{d_o}_o) \right]', \]

and matrices
\[ \Phi := \begin{bmatrix} \sum_{i=1}^{\tau} \rho(H(a^1_x, x^i_o), 0) & \cdots & \sum_{i=1}^{\tau} \rho(H(a^\kappa_x, x^i_o), 0) \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{\tau} \rho(H(a^1_x, x^i_o), d_o) & \cdots & \sum_{i=1}^{\tau} \rho(H(a^\kappa_x, x^i_o), d_o) \end{bmatrix}, \Phi_o := \begin{bmatrix} \sum_{i=1}^{\tau} \rho(H(x^1_o, x^i_o), 0) & \cdots & \sum_{i=1}^{\tau} \rho(H(x^\kappa_x, x^i_o), 0) \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{\tau} \rho(H(x^1_o, x^i_o), d_o) & \cdots & \sum_{i=1}^{\tau} \rho(H(x^\kappa_x, x^i_o), d_o) \end{bmatrix}. \]

The compact form representation of the optimization problem facilitates the computation of the equilibrium, e.g., via using linear programming (LP) methods, however, the dimension of the vector \( \mathbf{\alpha} \) can lead to computational issues for long codewords, i.e., large \( n \), even though the quotient set \( \mathcal{A}^d/\sim \) has reduced the size of \( \Lambda \)'s action space, without loss of generality, as shown in Lemma 1. To this end, in the following, we relax the attack space at large scales to reduce the size of the problem further substantially based on the equivalence relation (23).

C. Relaxing Attack Space at Large Scales

A codeword consists of the message and redundantly added symbols:
\[ \begin{bmatrix} \text{message} \\ \text{redundant symbols} \end{bmatrix} \in \Sigma^n \times \Sigma^m. \] (25)

Over \( \Sigma^n \), any two different encoded codewords have at most \( n \) and at least \( d \) different symbols. Therefore, the distance between any two encoded codewords can be in-between \( d \) to \( n \). Furthermore, the minimum distance between an arbitrary codeword and encoded codewords can be at most \( n - k \) since even though that codeword is completely different from all the encoded codewords at the redundant symbols, the message part must have matched with at least one encoded codeword. Therefore, we obtain
\[ \min H_o(a_x) \leq n - k \ \forall a_x \in \Sigma^n. \] (26)
Therefore, the reward for distances starting from the closest one $\min_j r_{\tau}$, where $r_{\tau}$ at the distances $d_n$ for each $\alpha_i$.

Furthermore, if $a_x \in \Sigma^n$ is in a decodable region of an encoded codeword, then there exists only that encoded codeword within $d - \min H_o(a_x)$ diameter.

At large scales, the number of messages $q^k$, i.e., the number of encoded codewords, is significantly larger than the length of the codewords $n$. We suppose that if $\min H_o(a_x) \leq d_o$, then there exists at least one encoded codeword at the distances $d - \min H_o(a_x), \ldots, n$. Otherwise, i.e., if $H_o(a_x) > d_o$, there exists at least one codeword at all the distances starting from the closest one $\min H_o(a_x)$ to $n$. In particular, formally, we suppose that

$$\{ \max\{d - \min H_o(a_x), \min H_o(a_x), \ldots, n\} \in H_o(a_x) \}. \quad (27)$$

Therefore, the reward for $a_x \in \Sigma^n$ depends mainly on the distance to the closest encoded codeword. For $i = 0, \ldots, n - k$, we define

$$r_i := \min_{\delta \in \{0, \ldots, n\}} \delta + \gamma \sum_{k=0}^{d} \rho(\delta, k) - \gamma \sum_{k=0}^{n} \rho(\delta, k) \quad \text{s.t. } \delta = i \lor \delta \geq \max\{d - i, i + 1\}, \quad (28)$$

where $r_i$, for $i = 0, \ldots, \nu$, denotes the reward when $\min H_o(a_x) = i$.

For each $a_x \in \Sigma^n$, $i = 1, \ldots, \kappa$, we introduce a $n + 1$ dimensional vector $\delta_i$ whose $j$th entry corresponds to the number of encoded codewords at distance $j - 1$. Then, $\Phi \in \mathbb{R}^{(d+1) \times \kappa}$ can be written as

$$\Phi = \begin{bmatrix} \rho(0, 0) & \cdots & \rho(n, 0) \\ \vdots & \ddots & \vdots \\ \rho(0, d_o) & \cdots & \rho(n, d_o) \end{bmatrix} \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_{\kappa} \end{bmatrix}. \quad (29)$$

Note also that all the entries of $\delta_i \in \mathbb{Z}^{n+1}$, $i = 1, \ldots, \kappa$, are non-negative integers and sum to the number of all encoded codewords $\tau$. However, these are necessary, yet not necessarily sufficient conditions, e.g., (26). Depending on $\min H_o(a_x)$, we can categorize $\delta_i$’s into the following $n - k + 1$ groups under two cases:

- $a_x \in \Sigma^n$ is in a decodable region, i.e., $\min H_o(a_x) \leq d_o$. Then, all the entries from the 1st to $(d - \min H_o(a_x))^\text{th}$ are zero except the one corresponding to $\min H_o(a_x)$, which is 1.
- $a_x \in \Sigma^n$ is not in any decodable region, i.e., $\min H_o(a_x) > d_o$. Then, all the entries from the 1st to $(\min H_o(a_x))^\text{th}$ are zero.

\(^2\)We note the index shift since $j$th entry corresponds to the number of encoded codewords at distance $j - 1$ rather than $j$. 
The unspecified entries of $\delta_i$ may not necessarily take arbitrary values; however, we will relax this and suppose that the unspecified entries can be set to arbitrary values by $A$ as long as they are all non-negative and all entries sum to $\tau$. The following $(n+1) \times (n-k+1)$ matrix concatenates these groups as an illustration:

$$
\begin{bmatrix}
0 & 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\
1 & 0 & 1 & \ldots & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
d_o & 0 & 0 & \ldots & 1 & 0 & \ldots & 0 \\
d_o + 1 & 0 & 0 & \ldots & * & * & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
n-k & 0 & * & \ldots & * & * & \ldots & * \\
d = n-k+1 & * & * & \ldots & * & * & \ldots & * \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
n & * & * & \ldots & * & * & \ldots & * \\
\end{bmatrix},
$$

where the entries denoted by * corresponds to arbitrary values, which can be at least 1 for consistency with (27). As an example, the first column corresponds to $a_x$ whose $\min H_{d_o}(a_x) = 0$, which yields that the second closest encoded codeword can be as close as $d - \min H_{d_o}(a_x) = d$.

Based on the relaxation that the unspecified entries can take any values, instead of a mixed strategy over $A^d / \sim$, we can consider a mixed strategy $\beta \in \Delta^n$, where

$$
\nu := (n - d + 1)(n - k + 1) + d_o(d_o + 1),
$$

as an approximation as shown in (32). Furthermore, we have

$$
r'\alpha = \tilde{r}'\beta,
$$

where

$$
\tilde{r}' := \begin{bmatrix} r_0 & \ldots & r_{n-k} \end{bmatrix}' \begin{bmatrix} 1' \\
\ddots \\
1' \end{bmatrix}
$$

and $r_i, i = 0, \ldots, n - k$, are given by (28). Therefore, we can write $D$’s (approximate) cost function in a similar compact form as

$$
\gamma_{\Delta}^D 1' \tilde{\Phi} \beta - \tilde{r}' \beta - \gamma_{\Delta}^D \begin{bmatrix} 0 \\
\pi' \end{bmatrix} \tilde{\Phi} \beta + \gamma_{\Delta}^F \begin{bmatrix} 0 \\
\pi' \end{bmatrix} \tilde{\Phi} \sigma_o + \xi,
$$

where, for notational simplicity, we define

$$
\tilde{\Phi} := \begin{bmatrix} \rho(0,0) & \ldots & \rho(n,0) \\
\vdots & \ddots & \vdots \\
\rho(0,d_o) & \ldots & \rho(n,d_o) \end{bmatrix} \Lambda.
$$

Note that in (35), $D$ selects $\pi \in [0,1]^d_o$ while $A$ selects $\beta \in \Delta^n$. Next, our goal is to transform $D$’s action into an action over a simplex at a higher dimensional space. To this end, we can view $\pi \in [0,1]^d_o$ as $D$ selects $d_o$ mixing strategies over two element sets, e.g., $\{0,1\}$. This yields that $D$ selects a mixed strategy over the Cartesian product space of these sets, i.e., $\times_{i=1}^{d_o} \{0,1\}$, which is

$$
\mu = 2^{d_o}
$$

dimensional. For example, for $d_o = 2$, the corresponding mixed strategy, denoted by $\hat{\pi} \in \Delta^\mu$, is over

$$
\{(1,0,1,0)', [1,0,0,1]', [0,1,1,0]', [0,1,0,1]'\}.
$$

This yields that

$$
\pi = \begin{bmatrix} 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 \end{bmatrix} \hat{\pi}.
$$
All the optimization arguments are now constrained over certain simplices. Since $1'\pi = 1$ and $1'\beta = 1$, we can write $\mathcal{D}$’s cost function as
\[
\min_{\pi} \max_{\beta} \pi' \Xi \beta + \xi, \tag{38}
\]
where $\Xi \in \mathbb{R}^{n \times n}$ is defined by
\[
\Xi := -1 \mathbf{r}' + \gamma D \mathbf{H}' (11' - \Pi' [0 \ I]) \Phi + \gamma F \Pi' [0 \ I] \Phi, \tag{40}
\]
where $\epsilon > 0$ and $\xi_o$ is the minimum entry of $\Xi$. Then, $\pi^*$ is given by \[19\]
\[
\pi^* = \Pi' \omega^* \hat{\Xi}^{-1} \omega^*. \tag{41}
\]
Remark. Note that by definition, we have
\[
\min_{\pi} \max_{\beta} \pi' \Xi \beta \geq \max_{\beta} \min_{\pi} \pi' \Xi \beta. \tag{42}
\]
Even though we are interested in only the upper value (38), the minimax theorem [19] shows that
\[
\min_{\pi} \max_{\beta} \pi' \Xi \beta = \min_{\beta} \max_{\pi} \pi' \Xi \beta, \tag{43}
\]
which implies that the upper and lower values of the game are equal and that we have a saddle-point equilibrium in (35). Note that (35) is obtained by relaxing $\Lambda$’s action space in (13), which is strategically equivalent to the original (11). And such interchangeability in the strategically equivalent zero-sum game does not imply interchangeability in the original nonzero-sum Stackelberg game [19]. \[\triangle\]

D. Transforming (38) to LP

We note that there exists a rather routine transformation of mixed-strategy equilibrium of zero-sum matrix games into an LP [19]. And we can view (38) as mixed-strategy equilibrium of a zero-sum matrix game. A sketch of the routine transformation of (38) into an LP [19] is as follows: i) we show that the game (38) is strategically equivalent to a game where the game matrix is a positive matrix, i.e., whose all entries are positive; ii) we can write (38) as the minimization of $\Lambda$’s best response; iii) we can obtain a certain necessary condition on $\pi$ in terms of $\Lambda$’s best response based on that $\Delta^\nu$ is a simplex; iv) through a change of variable, we can obtain an equivalent LP, which is given by
\[
\max 1' \omega \text{ subject to } \Xi ' \omega \leq 1, \omega \geq 0, \tag{39}
\]
where the positive matrix $\Xi_+ \in \mathbb{R}^{n \times n}$ is defined by
\[
\Xi_+ := \begin{cases} \Xi & \text{if } \Xi \text{ is a positive matrix} \\ \Xi + (\epsilon - \xi_o)11' & \text{otherwise}, \end{cases}
\]
where $\epsilon > 0$ and $\xi_o$ is the minimum entry of $\Xi$. Then, $\pi^*$ is given by [19]
\[
\pi^* = \Pi' \omega^* \hat{\Xi}^{-1} \omega^*. \tag{41}
\]
Remark. Note that by definition, we have
\[
\min_{\pi} \max_{\beta} \pi' \Xi \beta \geq \max_{\beta} \min_{\pi} \pi' \Xi \beta. \tag{42}
\]
Even though we are interested in only the upper value (38), the minimax theorem [19] shows that
\[
\min_{\pi} \max_{\beta} \pi' \Xi \beta = \min_{\beta} \max_{\pi} \pi' \Xi \beta, \tag{43}
\]
which implies that the upper and lower values of the game are equal and that we have a saddle-point equilibrium in (35). Note that (35) is obtained by relaxing $\Lambda$’s action space in (13), which is strategically equivalent to the original (11). And such interchangeability in the strategically equivalent zero-sum game does not imply interchangeability in the original nonzero-sum Stackelberg game [19]. \[\triangle\]

V. ILLUSTRATIVE EXAMPLES

As illustrative examples, we examine the performance of the proposed scheme with $[7,3,5]_8$ and $[15,3,13]_{16}$ Reed-Solomon codes for different symbol error probabilities, e.g., $p_e = .01, .1, .2$. Note that $[7,3,5]_8$ and $[15,3,13]_{16}$ can encode 512 (and 4096) traffic signs. The numbers of bits in a codeword, i.e., $n \times q$, are 56 and 240 for $[7,3,5]_8$ and $[15,3,13]_{16}$, respectively. We set all the traffic-signs equally likely. We use the Monte Carlo method to compute $\rho(\cdot, \cdot)$ over $10^6$ independent trials. In the following, we examine the performance over various scenarios in detail.

In Scenario-i), we consider relatively smaller set of traffic signs, e.g., $\tau = 512$. We use the $[7,3,5]_8$ code. Correspondingly, we have $d_o = 2, \mu = 4$ and $\nu = 21$. We examine the performance of the proposed scheme for the symbol error probabilities $p_e = .01$ and $p_e = .1$. In Fig. 4, we plot $\rho(\cdot, \cdot)$ for these symbol error probabilities. Recall
that $\rho(\delta, \delta_\circ)$ gives the probability of the distance between two codewords $x^1, x^2$, where $H(x^1, x^2) = \delta$, becomes $\delta_\circ$ after $x^2$ has been perturbed randomly by nature. We set

$$\gamma^L_\Lambda = \gamma^D_\Lambda = \gamma^F = 10. \tag{44}$$

In Table 4 we provide the optimal $\pi^* \in [0, 1]^d_\circ$. On the other hand, $\beta^* \in \Delta^\nu$ places positive weight only on $\min H_0(a_x) = 0$ while $H(a_t, a_x) = d = 5$, which implies that $\Lambda$ changes the codeword to a different closest one completely. We note that $\mathbb{D}$ achieves this level security at the expense of security and the proposed scheme can compute the optimal defense with respect to that trade-off.

In Scenario-ii), we consider relatively larger set of traffic-signs, e.g., $\tau = 4096$. To this end, we use the $[15, 3, 13]_{16}$ code. Correspondingly, we have $d_\circ = 6$, $\mu = 64$, and $\nu = 81$. This code leads to relatively longer codeword and correspondingly can correct more symbol errors. Therefore, we examine the performance of the proposed scheme for the symbol error probabilities $p_e = .1$ and $p_e = .2$. In Fig. 5 we plot $\rho(\cdot, \cdot)$ for these symbol error probabilities. We set

$$\gamma^L_\Lambda = \gamma^D_\Lambda = \gamma^F = 20. \tag{45}$$

Fig. 4: Probability of change of distance due to random noise for the $[7, 3, 5]_8$ code and $p_e = .01$ and $p_e = .1$. Note that scale of the color changes among plots.

Fig. 5: Probability of change of distance due to random noise for the $[15, 3, 13]_{16}$ code and $p_e = .1$ and $p_e = .2$. Note that scale of the color changes among plots.
TABLE I: Optimal defense strategy for Scenario-i for different symbol error probabilities while the best action is to change the symbol to a different closest one completely.

|   | $p_e = .01$ | $p_e = .1$ |
|---|---|---|
| $\pi^*(1)$ | .6860 | .1321 |
| $\pi^*(2)$ | .8605 | 1.000 |

TABLE II: Optimal defense strategy for Scenario-ii for different symbol error probabilities while the best action is to change the symbol to a different closest one completely.

|   | $p_e = .1$ | $p_e = .2$ |
|---|---|---|
| $\pi^*(1)$ | .1523 | .2239 |
| $\pi^*(2)$ | .1786 | .1650 |
| $\pi^*(3)$ | .2479 | .1613 |
| $\pi^*(4)$ | .5094 | .2282 |
| $\pi^*(5)$ | .8984 | .9552 |
| $\pi^*(6)$ | .9954 | .9965 |

In Table II we provide the optimal $\pi^* \in [0, 1]^d$. Again, we observe that $\beta^* \in \Delta^\nu$ places positive weight only on $\min H_0(a_x) = 0$ while now $H(g(a_x), a_x) = d - 1 = d = 13$, which also implies that $A$ changes the codeword to a different closest one completely.

VI. Conclusion

We have introduced a game theoretical error-correction framework to design classification algorithms that are reliable even in adversarial environments, with a specific focus on traffic-sign classification. Based on the analogy with a communication setting, we have proposed to incorporate the powerful error-correction methods to provide reliable and timely performance in classification. Different from general classification settings, traffic-sign classification is a suitable use-case to redesign the inputs physically by also considering intelligent vehicle systems. We have also analyzed the scheme within a game theoretical framework in order to design the system configurations optimally in terms of the trade-off between the false alarm and misclassification costs while the advanced attacker would be designing the attack optimally against the proposed defense. We have also provided an efficient method to mitigate possible computational issues that might arise at large scale environments, e.g., relatively more traffic-signs. We have transformed the problem into an LP with considerably small dimensions compared to the entire input space. Finally, we have examined the performance over various scenarios.

Some future research directions on this topic include the implementation of the idea on other classification algorithms. This approach can be a good fit for any safety- and time-critical system in which the classification can be viewed like a signaling problem. For example, computer vision for (warehouse) inventory management [31] or intelligent robotic sorting [32] can be other interesting applications for the framework developed here.

REFERENCES

[1] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, “Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey,” IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1484–1497, 2012.
[2] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, “Robust physical-world attacks on deep learning visual classification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[3] J. Jin, K. Fu, and C. Zhang, “Traffic sign recognition with hinge loss trained convolutional neural networks,” IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 1991–2000, 2014.
[4] A. González, L. M. Bergasa, and J. J. Yebes, “Text detection and recognition on traffic panels from street-level imagery using visual appearance,” IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 1, pp. 228–238, 2014.
[5] J. Greenhalgh and M. Mirmehdi, “Recognizing text-based traffic signs,” IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 3, pp. 1360–1369, 2015.
[6] Y. Yang, H. Luo, H. Xu, and F. Wu, “Towards real-time traffic sign detection and classification,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 7, pp. 2022–2031, 2016.
[7] Y. Zeng, X. Xu, D. Shen, Y. Fang, and Z. Xiao, “Traffic sign recognition using kernel extreme learning machines with deep perceptual features,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 6, pp. 1647–1653, 2017.
[8] C. Liu, F. Chang, and Z. Chen, “Rapid multiclass traffic sign detection in high-resolution images,” IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 6, pp. 2394–2403, 2014.
[9] X. Lu, Y. Wang, X. Zhou, Z. Zhang, and Z. Ling, “Traffic sign recognition via multi-modal tree-structure embedded multi-task learning,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 4, pp. 960–972, 2017.
[10] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” in arXiv:1312.6199, 2014.
[11] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in arXiv:1412.6572, 2015.
[12] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in 35th International Conference on Machine Learning (ICML), 2018.
[13] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, “Synthesizing robust adversarial examples,” in 35th International Conference on Machine Learning (ICML), 2018.
[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in 6th International Conference on Learning Representations, 2018.
[15] A. Raghunathan, J. Steinhardt, and P. Liang, “Certified defenses against adversarial examples,” in 6th International Conference on Learning Representations, 2018.
[16] H. Hosseini and R. Poovendran, “Semantic adversarial examples,” in arXiv:1804.00499, 2018.
[17] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, “NO need to worry about adversarial examples in object detection in autonomous vehicles,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[18] R. B. Myerson, Game Theory: Analysis of Conflict. Harvard University Press, 1997.
[19] T. Başar and G. Olsder, Dynamic Noncooperative Game Theory. Society for Industrial Mathematics (SIAM) Series in Classics in Applied Mathematics, 1999.
[20] N. Dalvi, P. Domingos, Mausam, S. Sanghavi, and D. Verma, “Adversarial classification,” in Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004.
[21] M. Brückner and T. Scheffer, “Nash equilibria of static prediction games,” in Proceedings of Advances in Neural Information Processing (NIPS), 2009.
[22] ———, “Stackelberg games for adversarial prediction problems,” in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011.
[23] L. Dritsoula, P. Loiseau, and J. Musacchio, “A game-theoretic analysis of adversarial classification,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 12, pp. 3094–3109, 2017.
[24] T. Ben-Bassat and D. Shinar, “Ergonomic guidelines for traffic sign design increase sign comprehension,” Human Factors, vol. 48, no. 1, pp. 182–195, 2006.
[25] R. H. Blahut, Algebraic Codes for Data Transmission. Cambridge University Press, 2002.
[26] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” Journal of the Society for Industrial and Applied Mathematics (SIAM), vol. 8, no. 2, pp. 300–304, 1960.
[27] A. Amini, R. M. Vaghefi, J. M. de la Garza, and R. M. Buehrer, “Improving GPS-based vehicle positioning for intelligent transportation systems,” in IEEE Intelligent Vehicles Symposium Proceedings, 2014.
[28] A.-G. A. Daraiseh and C. W. Baum, “Decoder error and failure probabilities for Reed-Solomon codes: Decodable vectors method,” IEEE Transactions on Communications, vol. 46, no. 7, pp. 857–859, 1998.
[29] E. Ok, Real Analysis with Economics Applications. Princeton University Press, 2007.
[30] D. P. Kroese, T. Brereton, T. Taimre, and Z. I. Botev, “Why the Monte Carlo method is so important today,” Wiley Interdisciplinary Reviews: Computational Statistics, vol. 6, no. 6, pp. 386–392, 2014.
[31] K. K. Katircioğlu and Y. Li, “Machine vision technology for shelf inventory management,” U.S. Patent US 2015/0 262 116 A1, Sep., 2015.
[32] J. Guerin, S. Thiery, E. Nyiri, and O. Gibaru, “Unsupervised robotic sorting: Towards autonomous decision making robots,” International Journal of Artificial Intelligence and Applications, vol. 9, no. 2, 2018.