Rare-Event Simulation for Neural Network and Random Forest Predictors

YUANLU BAI, Columbia University, USA
ZHIYUAN HUANG, Tongji University, China
HENRY LAM, Columbia University, USA
DING ZHAO, Carnegie Mellon University, USA

We study rare-event simulation for a class of problems where the target hitting sets of interest are defined via modern machine learning tools such as neural networks and random forests. This problem is motivated from fast emerging studies on the safety evaluation of intelligent systems, robustness quantification of learning models, and other potential applications to large-scale simulation in which machine learning tools can be used to approximate complex rare-event set boundaries. We investigate an importance sampling scheme that integrates the dominating point machinery in large deviations and sequential mixed integer programming to locate the underlying dominating points. Our approach works for a range of neural network architectures including fully connected layers, rectified linear units, normalization, pooling and convolutional layers, and random forests built from standard decision trees. We provide efficiency guarantees and numerical demonstration of our approach using a classification model in the UCI Machine Learning Repository.

CCS Concepts: • **Computing methodologies** → **Simulation theory**; **Simulation types and techniques**; **Machine learning**; • **Mathematics of computing** → **Probability and statistics**; **Mathematical optimization**.

Additional Key Words and Phrases: variance reduction, importance sampling, safety evaluation, neural network, random forest, large deviations

ACM Reference Format:
Yuanlu Bai, Zhiyuan Huang, Henry Lam, and Ding Zhao. 2022. Rare-Event Simulation for Neural Network and Random Forest Predictors. In . ACM, New York, NY, USA, 34 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Due to the extensive development of artificial intelligence (AI), machine learning techniques have been embedded in many safety-sensitive physical systems, including autonomous vehicles [79] and unmanned aircraft [77]. In autonomous vehicles, for instance, machine learning predictors can be applied to many tasks including perception [34, 114], path planning [50, 122], motion control [109], or end-to-end driving systems [35, 75, 87]. In these tasks, misprediction can cause catastrophic impacts on public safety, as exemplified by the series of fatal accidents encountered by autonomous driving systems due to the failures in detecting nearby vehicles or pedestrians (e.g. [21, 22]). To reduce the risk of such catastrophe, machine learning models in these systems need to be carefully evaluated against safety, especially before their mass deployment in public.

Recent research considers using probabilistic measures to quantify the risks of machine learning predictors or entire intelligent physical systems. These measures can be defined in a variety of ways. In *robustness evaluation*, a prediction model, with neural network as a dominant example, is considered more robust if it is more likely to make a consistent prediction under small perturbations on the input [60]. When the perturbation is modeled via a random distribution,

Manuscript submitted to ACM

the robustness of neural networks is measured by the probability that the prediction value persists [118–120]. In more complex *intelligent system evaluation*, risks can be quantified by the occurrence probabilities of safety-critical events. These events can be defined as the violation in terms of certain safety metrics (e.g., [48] listed seven potential safety metrics for autonomous vehicles including crashes per driving hour and disengagements per scenario), and recent studies use the probabilities of crash or injury in driving tasks as safety metrics [69, 91, 123, 124]. For AI-equipped autonomous vehicles, the evaluation target would implicitly involve a probabilistic measurement on the embedded machine learning model. Moreover, in [121], neural networks are further used to approximate sophisticated safety-critical sets defined from complex system dynamics, and the target probabilities comprise hitting sets defined via these neural network outputs.

Our study is motivated from the estimation of probabilistic risk measures described above. Due to the complexity of machine learning predictors, these probabilities are typically unamenable to analytical formulas, even when the underlying stochastic distribution is fully modeled. This thus calls for the use of Monte Carlo simulation. However, the target probabilities, which signify the risks of dangerous yet unlikely events, are tiny. The problem thus falls into the domain of rare-event simulation, in which it is widely known that crude Monte Carlo can be extremely inefficient and variance reduction is necessarily employed. Traditionally, rare-event simulation techniques (e.g. [31, 74]) have been applied in broad application areas including queueing systems [14, 15, 18, 45, 80, 98, 103, 110], communication networks [33, 76, 94], finance [52, 55, 56], insurance [5, 8, 37], reliability [67, 89, 90, 99, 112], biological processes [62, 105], dynamical systems [46, 116], and combinatorics [11, 13]. The evaluation of machine learning models and intelligent physical systems that we focus on here is a new application that is propelled rapidly by the growth of AI. Our goal is to provide a first step into building rare-event simulation algorithms in these applications, which integrate tools from both the disciplines of machine learning and rare-event simulation, and which are statistically guaranteed in terms of the classical efficiency notions in the rare-event literature.

More specifically, we study importance sampling (IS) [108] to design efficient estimators. In rare-event estimation, the rarity nature of hitting set dictates that crude Monte Carlo samples have a low frequency of observing the hitting occurrence, and this inefficiency exhibits statistically as a large relative error (i.e., ratio of standard deviation to mean) in the estimation. To mitigate this issue, IS uses an alternate distribution to generate samples that can attain a higher frequency in hitting the target event, and reweights the outputs to maintain unbiasedness via the likelihood ratios. To achieve a small relative error, the new generating distribution (i.e., the IS distribution) is carefully selected, often by analyzing the weights in interaction with the hitting set geometry and the underlying system dynamics [58, 104]. In this paper, we follow the above analysis path in the literature and use the common theoretical notion of efficiency called asymptotic optimality or logarithmic efficiency [9, 67, 74] that we will detail in the sequel.

In terms of our scope of study, we focus on piecewise linear machine learning predictors, which include random forests and neural networks with common activation functions such as rectified linear units (ReLU). The former is an ensemble or weighted average of decision trees [30], and the latter is a network of neurons connected in multiple layers, via the activation functions [59]. We also assume the underlying distribution is Gaussian or mixtures of such. Under this setting, we design provably efficient IS schemes to estimate rare-event probabilities that the prediction outputs hit above certain high thresholds. We will describe how our considered setup relates to the risk quantification of AI-driven algorithms or intelligent physical systems presented earlier, where our proposed approach provides a rigorous first step towards the resulting rare-event simulation problems (see Section 3).

Our main methodology integrates the classical notion of dominating points for rare-event sets with sequential mixed integer programming (MIP) to attain an efficient estimator. The notion of dominating points, and the associated

mixture-based IS scheme, is well-known in the literature [42, 104]. The MIP, while conceptually straightforward, requires leveraging recent formulations catered for the considered machine learning models. Let us explain the roles of these tools. Intuitively, a dominating point is the highest-density point in the rare-event set, so that using an IS distribution that shifts the mean to this point (via exponential tilting) gives rise to a distribution that hits the rare-event set more frequently, and the generated likelihood ratio contributes properly to the probability of interest, which are desirable for controlling the relative error. However, this is only a local characterization, as the simulation randomness could cause huge likelihood ratios for some generated samples. Controlling these ratios in turn requires a geometric property that, in the Gaussian case, implies the dominating point to be on the boundary of the rare-event set, and that the latter lies completely inside one of the half-spaces cut by the tangential hyperplane passing through the dominating point (e.g., these occur when the rare-event set is convex). When this geometric property does not hold, then one needs to divide the rare-event set into a union of smaller sets each bearing its own dominating point, and an efficient IS scheme is built via a mixture of exponential tiltings targeted at all these individual dominating points [104]. The sequential MIP in our procedure serves to locate all these dominating points. It casts the search as a density maximization problem constrained by hitting sets induced from the considered machine learning model. The involved feasible regions shrink sequentially as we add more "cutting planes" to the constraints in order to remove the half-spaces that are already considered by earlier dominating points. Our MIPs are derived from the reformulation techniques that appeared recently in the machine learning literature, which leverage the geometric structures of ReLU neural networks [111] and random forests [86]. We provide a step-by-step guide in formulating random forests and different neural network architectures as suitable MIPs to be inserted into our sequential algorithm.

In terms of theoretical results, we show asymptotic optimality of our IS that targets at general piecewise polyhedrons, which apply to our considered rare-event sets in particular. Towards this, we also derive large deviations results for the associated probabilities of interest. Our results are developed under a different regime from the conventional one in the literature. More specifically, the latter typically scales the input random vector that falls into a fixed set (e.g. [68, 104]), while we let the exceedance threshold on the output of the machine learning model to scale. Our setting is more natural since the threshold provides meaning in defining the level of risk (e.g., in vehicle safety test, the relative velocity at the crash time can be used to compute a so-called Maximum Abbreviated Injury Score that predicts the severity of injuries [83], and hence the probabilities of relative velocity at the crash time exceeding different thresholds are of interest). To this end, the closest work that studies a similar regime is [64], but it only analyzes the tail probability that a Gaussian random vector is componentwise larger than a threshold, which is a simplified version of our regime without the machine learning transformation. While we leverage the results in [64], we also develop mathematical techniques to make the generalization fit our setting.

The paper is organized as follows. Section 2 first provides a literature review. Section 3 describes and motivates our problem setting. Section 4 presents our algorithm and theoretical guarantees. Section 5 provides the MIP formulations for random forests and different neural network architectures. Section 6 shows numerical results. Section 7 contains the proofs of theorems.

2 RELATED WORK

A significant line of work studies the use of large deviations to invent efficient IS procedures, which mathematically identifies the most likely path to trigger a rare event through minimizing the so-called rate function (see, e.g., the surveys [9, 17, 31, 53, 74, 102]). This approach leads to the concept of dominating points and mixture IS [42, 104] which our work follows. Despite this utilization, our work differs from the previous works. First is that our considered machine

learning models, including random forests and neural networks, deem the rare-event boundaries to be only expressible implicitly. This in turn necessitates the use of sequential MIP algorithm that can leverage such expressions in the search of the dominating points. This distinguishes our approach from [3, 92] that similarly consider splitting rare-event sets via dominating points, but constrain the rare-event sets to be unions of half-spaces that are explicitly given. Second, we derive asymptotic results for the rare-event probability of interest and show efficiency of our algorithm, as the exceedance threshold increases, a regime subtly different from the majority of literature yet more natural in our setting. To this end, [64] appears closest to our work, with derived bounds and asymptotic results for the tail probability of Gaussian random vectors. However, our setting is considerably more complex as it involves piecewise linear machine learning predictor output, and correspondingly requires more intricate analysis coming from the geometry of the rare-event set. Next, similar to our derivations, [68] represents the asymptotic of probability on convex sets using dominating points, but they focus on a different scaling from ours. Specifically, like in standard large deviations theory, they focus on the conventional regime where the scaled componentwise maximum of Gaussian random vectors lies in a fixed convex set, while our target event is that the predictor output with Gaussian input (which is not scaled and cannot be expressed as a componentwise maximum) exceeds an increasing threshold.

In the machine learning literature, some studies use probabilistic measures to evaluate the robustness of prediction models. Since these measures can be extremely small, rare-event simulation techniques are considered. [119] discusses an adaptive multilevel splitting approach to estimate the statistical robustness of machine learning models. [113] considers the problem of estimating agent failure probabilities and proposes to learn a failure probability predictor to approximate the minimum-variance IS distribution. [120] proposes an approach to compute the lower and upper bounds for a probabilistic robustness measure. Our work is motivated by the topics studied in these works, and can be viewed as a step towards the provision of rigorous guarantees for methodologies driven by the corresponding applications.

Another related line of research studies optimization problems with machine learning models in the objective. [86] discusses the optimization of tree ensemble models and provides treatment for large scale problems. [111] formulates the robustness verification of neural networks as MIP problems. These studies leverage the piecewise linear property of these machine learning models to turn optimization on the prediction output into tractable MIPs. Our MIP formulations for finding dominating points follow from these optimization studies.

We close this literature review by briefly discussing other IS schemes. The cross-entropy method [24, 38, 96, 100, 101] uses sequential stochastic optimization to search for an optimal IS distribution in a parametric family. Adaptive IS [2, 26, 41, 78] updates the IS distribution iteratively between simulated replications to approach the optimal (zero-variance) IS distribution and generates non i.i.d samples for estimating the target expectation associated with finite-state discrete Markov chains. Another line of studies use techniques such as Markov-chain Monte Carlo (MCMC) to sample from the rare-event set of interest, or approximately from the conditional distribution given the occurrence of the rare event [27, 28, 32, 61]. IS schemes have also been designed for heavy-tailed systems [16, 19, 20, 33, 44, 72, 88], in contrast to the light-tailed settings considered in this paper. Besides IS, other competing methods for rare-event simulation include conditional Monte Carlo [6, 7] and splitting [39, 51, 54, 84, 93].

3 PROBLEM SETTING

We state our problem setting. Consider a prediction model $g(\cdot)$, with input $X \in \mathbb{R}^d$ and output $g(X) \in \mathbb{R}$. Suppose that the input follows a Gaussian distribution, i.e, $X \sim N(\mu, \Sigma)$, where Σ is a $d \times d$ positive definite matrix. We want to estimate the probability $p = P(X \in S)$, where $S = \{x : g(x) \geq \gamma\}$ is a rare-event set with a threshold $\gamma \in \mathbb{R}$ that triggers

the rare event. We note that the Gaussian assumption can be relaxed without much difficulty in our framework to, for instance, mixtures of Gaussians, which we will discuss later and can expand our scope of applicability.

This problem setting is motivated from risk assessments involving machine learning models, as exemplified below.

Example 3.1 (Statistical Robustness Metric). Studies on robustness of machine learning models have become increasingly prevalent in recent years. The topic was initiated in computer vision studies [60], where neural networks for image classification were found to be vulnerable to tiny perturbation to the input. Such a perturbed input is considered as an adversarial example. Studies have discussed how to find these adversarial examples [81] and to conduct adversarial learning [82] in more general machine learning tasks. The vulnerability to perturbation has caused safety and security concerns about using machine learning models in real-life applications. In order to evaluate how robust a prediction model is under potential perturbations, robustness metrics are proposed as quantitative benchmarks.

For instance, we consider a classification model $g(\cdot)$ that can correctly predict the input x_0 as category c. Intuitively, the model is "robust" at x_0 if the correct prediction remains for all x such that $d(x, x_0) \leq \epsilon$ where d denotes a certain distance and $\epsilon > 0$ is a small real number. Based on this intuition, a statistical robustness metric considers $p = P(g(X) \neq c)$, where X follows a distribution concentrated around x_0 [119, 120]. Here p represents the probability that the output is inconsistent with the baseline prediction at x_0.

In particular, when $g(\cdot)$ predicts using "score functions" $g_i(\cdot)$ with $i = 1, .., K$ where K denotes the number of categories, the predicted output is the category that has the maximum score, i.e. the prediction at x is given by $\arg\max_i g_i(x)$. Then we note that $g(x) \neq c$ is equivalent to $g_c(x) \leq \max_{i \neq c} g_i(x)$. Hence we can transform p into $P(\tilde{g}(X) \geq 0)$ by defining $\tilde{g}(x) = \max_{i \neq c} g_i(x) - g_c(x)$, which reduces to our problem statement presented earlier.

Example 3.2 (Risk Evaluation of Intelligent Physical Systems). Many intelligent physical systems (e.g. driver assistance systems) are built in a modular structure, which divides the overall task into sub-tasks that are handled by different modules. The perception module extracts information from the environment through various sensors (e.g. LIDAR [49], camera, etc.), which provides input for the downstream tasks [43]. Nowadays, perception modules are usually integrated with machine learning models, which play crucial roles in converting raw sensor data (e.g. images, point clouds) into information that are readable by downstream modules (e.g. object class, bounding box) [97].

Consider an intelligent physical system that embeds a machine learning predictor g for perception (e.g. object detection). We then represent the decision of the system given an input x as $h(g(x))$. The probability $P(h(g(X)) \in S)$, where S represents a risky region, can be used to measure the risk of the system decision. For instance, suppose we are evaluating a collision avoidance system via the probability of a severe injury. Here, x can represent the sensor data of a collision scenario, $h(g(x))$ the relative speed when collision happens, which proxies the severity of potential injuries, and the evaluation is equivalent to estimating $P(h(g(X)) \geq \gamma)$ for some speed threshold γ.

In most cases, h is random by itself and can have a different complexity structure than the function class g. Our setup, which drops the general random h, can be viewed as a simplified probability $P(g(X) \geq \gamma)$ that provides a first step of study along this direction.

Example 3.3 (Probability Evaluation for Learned Rare-Event Set). When the system that drives the rare event is a black-box or too complicated to analyze [40], an approach to retain tractability is to approximate or learn the rare-event set via machine learning tools [4, 121]. An example in operations research is the prediction of congestion risks in sophisticated queueing systems arising in, e.g., healthcare applications [29], where the queue could have multiple classes of customers and complex priority rules [63] and the event of interest could be a transient probability of high occupancy level. In such settings, we can collect collect data or run simulations for $\{X, Y\}$, where X denotes the random object

in the considered system and $Y \in \{0, 1\}$ denotes either the occurrence of the considered rare event or the numerical outcome under input X. Then we train neural network $g(\cdot)$ to classify the rare-event region given X. The learned rare-event set is then represented by $\{x : g(x) \geq \gamma\}$, where γ is the threshold for classifying rare-event (e.g. $\gamma = 0.5$) or the threshold for the outcome to trigger the event. As a result, $p = P(g(X) \geq \gamma)$ provides an approximation to the rare-event probability.

Example 3.4 (Validating Classification Models With Rare Categories). In classification model validation tasks, estimating the predictive performance of the test model can be costly if the test data requires human-annotation [107]. When we are interested in the performance on a rare category, the estimation of predictive metrics, e.g. F-scores (or F-measures) [115], becomes more challenging and hence requires more efficient approaches than naive sampling [95]. Consider that the input of the test classification model, denoted by X, has a fixed probability distribution across the population of samples. We use $y(X) \in \{1, 0\}$ to denote the correct annotation at the input X and $g(X)$ to denote the prediction given by the test classifier. Suppose we are interested in the prediction accuracy of the rare category $y(X) = 1$ (i.e. $P(y(X) = 1)$ is extremely small). The F_α-measure of the classification model is defined by:

$$F_\alpha = \frac{P(y(X) = 1, g(X) = 1)}{\alpha P(g(X) = 1) + (1 - \alpha) P(y(X) = 1)}, \tag{1}$$

with $\alpha \in (0, 1)$. We observe that when $y(X) = 1$ is a rare event and the classifier g is well trained, all three probabilities in the F_α-measure can be extremely small. Therefore, accurately estimating the F_α-measure is closely related to estimating the rare-event probabilities $P(y(X) = 1, g(X) = 1)$ and $P(g(X) = 1)$, which are defined via the test prediction model g.

Our setup described in the beginning of this section thus relates to the four emerging examples above. Though we could not resolve all the issues in these examples, notably with restrictions on the input distribution and model complexity, we view our study as a first step towards a rigorous use of rare-event simulation techniques developed among the stochastic simulation community in the surging domain of safety and risk evaluation of AI-driven systems.

4 EFFICIENT IMPORTANCE SAMPLING VIA SEQUENTIAL MIXED INTEGER PROGRAMMING

We present our IS methodology. Section 4.1 reviews IS basics. Section 4.2 describes how we integrate the notions of dominating points and mixture IS with a sequential MIP algorithm. Section 4.3 presents our theoretical efficiency guarantees. The reformulation and solution to the MIP algorithms, which utilize recent developments in machine learning, are discussed in Section 5.

4.1 Basics of Importance Sampling

When p is small, estimation using crude Monte Carlo is challenging since, intuitively, the samples have a low frequency of hitting the target set. This is statistically manifested as a large relative error. To be more specific, suppose that we use the crude Monte Carlo estimator $\hat{p}_N = \frac{1}{N} \sum_{i=1}^{N} I(g(X_i) \geq \gamma)$ to estimate p. Since the probability p is tiny, the error of the estimator should be measured relative to the size of p. In other words, we would like the probability of having a large relative error to be small, i.e., $P(|\hat{p}_N - p| > \varepsilon p) \leq \delta$ where δ is the confidence level and $0 < \varepsilon < 1$. By Markov's inequality, a sufficient condition for this is

$$N \geq \frac{Var(I(g(X) \geq \gamma))}{\delta \varepsilon^2 E[I(g(X) \geq \gamma)]^2} = \frac{RE^2}{\delta \epsilon^2}.$$

where $RE = \sqrt{Var(I(g(X) \geq \gamma))}/E[I(g(X) \geq \gamma)]$ is the relative error. For the crude Monte Carlo estimator, the RE is given by $\sqrt{(1-p)/p}$. That is, the simulation size N has to be roughly proportional to $1/p$ in order to achieve a given relative error. Under the settings that X has a Gaussian distribution and g is piecewise linear (see Corollary 4.3), p is exponentially small in the threshold level γ, and hence the required simulation size would grow exponentially in γ.

A common approach to speed up simulation in such contexts is to use IS (see, e.g. the surveys [9, 17, 31, 53, 74, 102], among others). Suppose X has a density f. The basic idea of IS is to change the sampling distribution to say $\tilde{f}$, and output

$$Z = I(g(\tilde{X}) \geq \gamma)\frac{f(\tilde{X})}{\tilde{f}(\tilde{X})}, \tag{2}$$

where $\tilde{X}$ is sampled from $\tilde{f}$. This output is unbiased if f is absolutely continuous with respect to $\tilde{f}$ over the rare-event set $\{x : g(x) \geq \gamma\}$ since

$$\tilde{E}[Z] = \int_{\mathbb{R}} I(g(x) \geq \gamma)\frac{f(x)}{\tilde{f}(x)}\tilde{f}(x)dx = \int_{\mathbb{R}} I(g(x) \geq \gamma)f(x)dx = E[I(g(X) \geq \gamma)] = P(g(X) \geq \gamma).$$

By choosing $\tilde{f}$ appropriately, one can substantially reduce the simulation variance.

To measure the efficiency of an IS scheme, we introduce a rarity parameter, say γ, that parametrizes the rare-event probability p_γ such that $p_\gamma \to 0$ as $\gamma \to \infty$. As discussed before, since the probability of interest is small, one should focus on the relative error of the Monte Carlo estimator with respect to the magnitude of this probability. To this end, we call an IS estimator Z_γ for p_γ asymptotically optimal [9, 74] if

$$\lim_{\gamma\to\infty} \frac{\log \tilde{E}[Z_\gamma^2]}{\log \tilde{E}[Z_\gamma]} = 2, \tag{3}$$

where $\tilde{E}$ denotes the expectation with regard to $\tilde{f}$. The notion (3) is equivalent to saying that $\tilde{E}[Z_\gamma^2]$ and $\tilde{E}[Z_\gamma]^2$ grow in the same exponential rate in γ. This ensures that the second moment, or the variance, does not explode exponentially relative to the probability of interest as γ increases, thus preventing an exponentially large number of simulation replications to achieve a given relative accuracy. We will use asymptotic optimality as our efficiency criterion in this paper. Moreover, in the large deviations settings where $p_\gamma = \tilde{E}[Z_\gamma]$ decays exponentially in γ, $\tilde{E}[Z_\gamma^2]/\tilde{E}[Z_\gamma]^2$ at most growing polynomially in γ is a sufficient condition for asymptotic optimality.

Another commonly used efficiency criterion is the bounded relative error, which is defined as

$$\limsup_{\gamma\to\infty} \frac{\tilde{E}[Z_\gamma^2]}{\tilde{E}[Z_\gamma]^2} < \infty.$$

This is a stronger condition than asymptotic optimality. More efficiency criteria can be found in [73, 85].

4.2 Dominating Points and Mixture Importance Samplers

In the case of Gaussian input distributions, finding a good $\tilde{f}$ is particularly handy and one approach to devise good IS distributions uses the notion of so-called dominating point. As explained in the introduction, a dominating point can be understood as the highest-density point in the rare-event set that satisfies some conditions. More precisely, the collection of dominating points for a rare-event set with Gaussian distributed input is defined in Definition 4.1.

Definition 4.1. Suppose that $S \subset \mathbb{R}^d$ is a rare-event set. Suppose that a set $A \subset \mathbb{R}^d$ satisfies that $S \subset \bigcup_{a\in A}\{x : (a-\mu)^T\Sigma^{-1}(x-a) \geq 0\}$ and that $a = \arg\min\{(x-\mu)^T\Sigma^{-1}(x-\mu) : x \in S \text{ and } (a-\mu)^T\Sigma^{-1}(x-a) \geq 0\}$ for any $a \in A$.

Moreover, suppose that the above conditions do not hold anymore if we remove any element from A. Then the points in A are called the dominating points of S with input distribution $N(\mu, \Sigma)$.

Note that minimizing $(x-\mu)^T\Sigma^{-1}(x-\mu)$ is equivalent to maximizing $\phi(x;\mu,\Sigma)$, the Gaussian density with mean μ and covariance Σ. The condition $2(a-\mu)^T\Sigma^{-1}(x-a) \geq 0$ is the first-order condition of optimality for the optimization $\min_x (x-\mu)^T\Sigma^{-1}(x-\mu)$ over a convex set for x. Thus, intuitively, each dominating point in the collection A can be viewed as the highest-density point in a "local" region formed by $S \cap \{x : (a-\mu)^T\Sigma^{-1}(x-a) \geq 0\}$. Figure 1 is an illustration of the dominating points. In particular, if $\{x : g(x) \geq \gamma\}$ is a convex set, then there is only one dominating point a. In this case, a well-known IS scheme is to use a Gaussian distribution $N(a, \Sigma)$ as the IS distribution $\tilde{f}$.

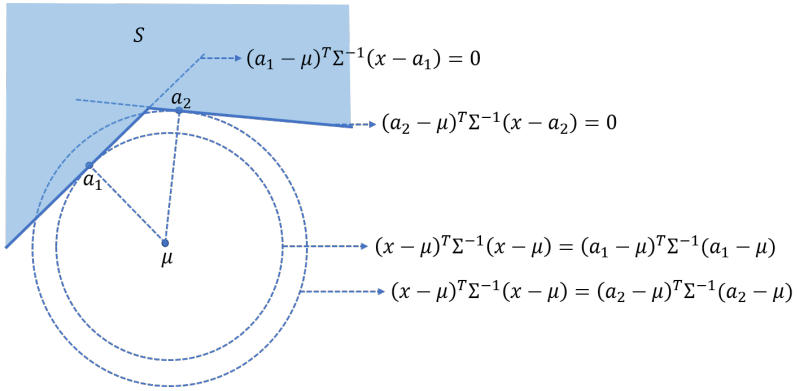

Fig. 1. Illustration of the dominating points. a_1 is the globally highest-density point in the rare-event set S, but the halfspace $\{x : (a_1-\mu)^T\Sigma^{-1}(x-a_1) \geq 0\}$ does not fully cover S, so an additional point a_2 is needed to comprise a dominating set.

We explain intuitively why we need more than one dominating point (the highest-density point over S) and the pitfall if we omit the other ones in constructing efficient IS. Suppose that the rare-event set consists of two disconnected convex components which are nearly equi-distant with respect to the origin, and we choose the IS distribution to be centered at the dominating point of one component. Then, if a sample from the IS distribution hits the other component, a scenario that could be unlikely but possible, the resulting likelihood ratio, which now contributes to the output as the rare-event set is hit, could possibly be tremendous. This ultimately leads to an explosion of the relative error in the IS estimator. [57] presents more counterexamples which show that it is essential to find all the dominating points in constructing an efficient IS based on mixtures.

In view of the aforementioned discussions, we consider the following IS scheme. If we can split $\{x : g(x) \geq \gamma\}$ into $\mathcal{R}_1, ..., \mathcal{R}_r$, and for each $\mathcal{R}_i, i = 1, ..., r$ there exists a dominating point a_i such that $a_i = \arg\min\{(x-\mu)^T\Sigma^{-1}(x-\mu) : x \in \mathcal{R}_i\}$ and $\mathcal{R}_i \subseteq \{x : (a_i-\mu)^T\Sigma^{-1}(x-a_i) \geq 0\}$, then we use a Gaussian mixture distribution with r components as the IS distribution $\tilde{f}$, where the ith component has mean a_i. This proposal guarantees the asymptotic optimality of the IS (see Theorem 4.2).

In our task, because the machine learning predictor $g(x)$ is nonlinear and x is high-dimensional in general, splitting $\{x : g(x) \geq \gamma\}$ into $\mathcal{R}_1, ..., \mathcal{R}_r$ that have dominating points is challenging even with known parameters. This challenge motivates us to use Algorithm 1 to obtain the dominating points $a_1, ..., a_r$ that constructs an efficient IS distribution. The procedure uses a sequential "cutting plane" approach to exhaustively look for all dominating points, by reducing the search space at each iteration via taking away the regions covered by found dominating points. The set A in the

procedure serves to store the dominating points we have located throughout the procedure. At the end of the procedure, we obtain a set A that contains all the dominating points $a_1, ..., a_r$. Note that when $g(x) \geq \gamma$ is convex, the algorithm solves a series of convex quadratic programming problems, and it is well known that such problems could be solved efficiently in polynomial time (see [117] for more details on the complexity). In this paper, we focus on the problems with piecewise linear $g(x)$, which leads to mixed integer convex quadratic optimization problems as shown in later discussion. Although a mixed integer quadratic optimization is NP-hard, we can solve it much more efficiently using specialized algorithms than general nonlinear MIPs [25].

Algorithm 1: Procedure to find all dominating points for the set $\{x : g(x) \geq \gamma\}$.

Input: Prediction model $g(x)$, threshold γ, input distribution $N(\mu, \Sigma)$.
Output: dominating point set A.

Start with $A = \emptyset$;
While $\{x : g(x) \geq \gamma, (a_i - \mu)^T \Sigma^{-1}(x - a_i) < 0, \ \forall a_i \in A\} \neq \emptyset$ **do**
    Find a dominating point a by solving the optimization problem

$$\begin{aligned} a = \arg\min_x \ & (x-\mu)^T\Sigma^{-1}(x-\mu) \\ s.t. \ & g(x) \geq \gamma \\ & (a_i-\mu)^T\Sigma^{-1}(x-a_i) < 0, \ \forall a_i \in A \end{aligned} \tag{4}$$

    and update $A \leftarrow A \cup \{a\}$;
End

Algorithm 1 gives $A = \{a_1, \ldots, a_r\}$. With this, we split $\{x : g(x) \geq \gamma\}$ into $\mathcal{R}_1, \ldots, \mathcal{R}_r$ where $\mathcal{R}_i = \{x : g(x) \geq \gamma, (a_i - \mu)^T\Sigma^{-1}(x - a_i) \geq 0, (a_j - \mu)^T\Sigma^{-1}(x - a_j) \leq 0, \forall j < i\}$. Clearly $a_i = \arg\min\{(x-\mu)^T\Sigma^{-1}(x-\mu) : x \in \mathcal{R}_i\}$ and $(a_1 - \mu)^T\Sigma^{-1}(a_1 - \mu) \leq \cdots \leq (a_r - \mu)^T\Sigma^{-1}(a_r - \mu)$. Moreover, we note that $(a_1 - \mu)^T\Sigma^{-1}(a_1 - \mu) = \min_{i=1,\ldots,r}\{(a_i - \mu)^T\Sigma^{-1}(a_i - \mu)\}$.

Given the dominating point set A, we use a mixture distribution with density

$$\tilde{f}(x) = \frac{1}{r}\sum_{i=1}^{r} \phi(x; a_i, \Sigma)$$

as the IS distribution. That is, the IS estimator is

$$Z = I(g(\tilde{X}) \geq \gamma)L(\tilde{X}) \tag{5}$$

where $\tilde{X} \sim \tilde{f}$ and L, the likelihood ratio, is defined as

$$L(x) = \frac{f(x)}{\tilde{f}(x)} = \frac{re^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}}{e^{-\frac{1}{2}(x-a_1)^T\Sigma^{-1}(x-a_1)} + \cdots + e^{-\frac{1}{2}(x-a_r)^T\Sigma^{-1}(x-a_r)}}.$$

Note that we have used uniform mixture weights in our IS distribution depicted above. These weights could potentially be tuned more carefully rather than simply equally assigned to further improve the efficiency, especially when an asymptotic zero-variance distribution is available (as in, e.g., [1, 66]), though here we are contented with uniform weights and do not refine further. To sum up, after computing the dominating points $A = \{a_1, \ldots, a_r\}$ using Algorithm 1, we estimate the probability of interest via Algorithm 2.

Algorithm 2: Construct the IS estimator with all the dominating points.

Input: Prediction model $g(x)$, threshold γ, dominating points $A = \{a_1, \ldots, a_r\}$, simulation size N.

Output: Estimated rare-event probability $\hat{p}$.

1 Generate $\tilde{X}_1, \ldots, \tilde{X}_N \sim \tilde{f}(x) = \frac{1}{r}\sum_{i=1}^{r} \phi(x; a_i, \Sigma)$;

2 Compute $\hat{p} = \frac{1}{N}\sum_{i=1}^{N} I(g(\tilde{X}_i) \geq \gamma) L(\tilde{X}_i)$ where

$$L(x) = \frac{r e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}}{e^{-\frac{1}{2}(x-a_1)^T\Sigma^{-1}(x-a_1)} + \cdots + e^{-\frac{1}{2}(x-a_r)^T\Sigma^{-1}(x-a_r)}};$$

3 **End**

4.3 Efficiency Guarantees

The efficiency guarantee of the proposed IS estimator (5) is given by:

Theorem 4.2. *Suppose that the input $X \sim N(\mu, \Sigma)$ and the prediction model $g(\cdot)$ is a piecewise linear function (with finite pieces) such that $P(g(X) \geq \gamma) > 0$ for any $\gamma \in \mathbb{R}$. The IS estimator Z is defined in (5). Then we have that $\tilde{E}[Z^2]/\tilde{E}[Z]^2$ is at most polynomially growing in γ. Moreover, Z is asymptotically optimal.*

Theorem 4.2 is proved by constructing an upper bound for the relative error, which in turn depends on the asymptotic approximation of probability on polytope sets using dominating points. Our proof leverages the results in [64] on the tail exceedance asymptotic of $P(N(0, \Sigma_n) \geq t_n)$ where $\|t_n\| \to \infty$ as $n \to \infty$, but requires substantial generalization. Note that Theorem 4.2 only makes the very general assumptions that g is piecewise linear and the probability $P(g(X) \geq \gamma)$ is nondegenerate (i.e., non-zero) for any $\gamma \in \mathbb{R}$. Our result applies to, for example, the probability $P(AX \geq t)$ where A is a constant matrix and $t - \gamma e_1$ is a constant vector (here, $e_1 = (1, 0, \ldots, 0)^T$). If AA^T is not invertible, then it is not easily reducible to the setting studied in [64]. To achieve a general result, we carefully construct a superset and a subset of the rare-event set to derive tight enough upper and lower bounds for the probability of interest, in which we analyze the involved asymptotic integrals instead of using the conditional probability representation in [64] that is not directly applicable in our setting. For the detailed proof, please refer to Section 7.

A by-product in deriving Theorem 4.2 is the large deviations probability asymptotic for $P(g(X) \geq \gamma)$:

Corollary 4.3. *Suppose that the input $X \sim N(\mu, \Sigma)$ and the prediction model $g(\cdot)$ is a piecewise linear function (with finite pieces) such that $P(g(X) \geq \gamma) > 0$ for any $\gamma \in \mathbb{R}$. Denote $a = \arg\min\{(x-\mu)^T\Sigma^{-1}(x-\mu) : g(x) \geq \gamma\}$. Then $-\log P(g(X) \geq \gamma) = (1 + o(1))(a-\mu)^T\Sigma^{-1}(a-\mu)/2$ as $\gamma \to \infty$. In particular, $P(g(X) \geq \gamma)$ is exponentially small in γ.*

The theoretical guarantee given by Theorem 4.2 justifies the sequential MIP algorithm for searching dominating points. The resulting mixture IS distribution is asymptotically optimal. We point out some related works that use mixture distributions that are related to our proposed method. In [3, 92], mixture IS distributions are constructed based on separating rare-event set with half-spaces. However, in these works, the rare-event set is restricted to be a union of half-spaces, and these half-spaces are assumed to be known. The use of Algorithm 1 allows us to deal with more general rare-event sets. Moreover, in relation to Corollary 4.3, we also mention the work [68] that derives an asymptotic result for Gaussian probabilities using dominating points. However, they focus on convex hitting sets where the entire set is scaled with a rarity parameter, which is different from our settings. First, our rare-event set is not necessarily convex. Second, even if we separate our rare-event set into the union of convex sets, their results still cannot be applied, since in our settings some linear constraints are allowed to be fixed instead of scaling with γ.

Finally, we close this section by noting that the proposed IS scheme can be extended to problems with Gaussian mixture inputs. Suppose the Gaussian mixture has m components, so that $X \sim \sum_{j=1}^{m} \pi_j \phi(x; \mu_j, \Sigma_j)$. For each component j, we implement Algorithm 1 with input distribution $N(\mu_j, \Sigma_j)$ to obtain dominating point set A_j (with cardinality r_j). The proposed IS distribution is given by $\tilde{f}(x) = \sum_{j=1}^{m} \sum_{i=1}^{r_j} \pi_j / r_j \phi(x; a_{ji}, \Sigma_j)$. We summarize the procedure as Algorithm 3.

Algorithm 3: Procedure for Gaussian mixture distributed input.

Input: Prediction model $g(x)$, threshold γ, input distribution $\sum_{j=1}^{m} \pi_j \phi(x; \mu_j, \Sigma_j)$, simulation size N.

Output: Estimated rare-event probability $\hat{p}$.

1 Implement Algorithm 1 with input distribution $N(\mu_j, \Sigma_j)$ to get $A_j = \{a_{j1}, \ldots, a_{jr_j}\}$;

2 Generate $\tilde{X}_1, \ldots, \tilde{X}_N \sim \tilde{f}(x) = \sum_{j=1}^{m} \sum_{i=1}^{r_j} \pi_j / r_j \phi(x; a_{ji}, \Sigma_j)$;

3 Compute $\hat{p} = \frac{1}{N} \sum_{i=1}^{N} I(g(\tilde{X}_i) \geq \gamma) L(\tilde{X}_i)$ where

$$L(x) = \frac{\sum_{j=1}^{m} \pi_j |\Sigma_j|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)}}{\sum_{j=1}^{m} \sum_{i=1}^{r_j} \pi_j / r_j |\Sigma_j|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-a_{ji})^T \Sigma_j^{-1}(x-a_{ji})}}; \tag{6}$$

4 **End**

Similar to Algorithm 2, we have the efficiency guarantee for Algorithm 3:

Corollary 4.4. *Suppose that the input $X \sim \sum_{j=1}^{m} \pi_j \phi(x; \mu_j, \Sigma_j)$ and the prediction model $g(\cdot)$ is a piecewise linear function (with finite pieces) such that $P(g(X) \geq \gamma) > 0$ for any $\gamma \in \mathbb{R}$. The IS estimator Z is defined as $I(g(\tilde{X}) \geq \gamma) L(\tilde{X})$ where $\tilde{X} \sim \sum_{j=1}^{m} \sum_{i=1}^{r_j} \pi_j / r_j \phi(x; a_{ji}, \Sigma_j)$ and $L(x)$ is as defined in (6). Then we have that $\tilde{E}[Z^2]/\tilde{E}[Z]^2$ is at most polynomially growing in γ. Moreovers, Z is asymptotically optimal.*

When we apply Algorithm 1 to find all dominating points, the key is to be able to solve the optimization problems in (4). We will investigate this in the next section.

5 TRACTABLE OPTIMIZATION FORMULATION FOR PREDICTION MODELS

We discuss how to formulate the optimization problems in Algorithm 1 as an MIP with quadratic objective function and linear constraints, for random forest (Section 5.1) and neural network (5.2) structures.

5.1 Tractable Formulation for Random Forest

A random forest [30, 65] can be specified as follows. Given a set of T decision trees $g_1, ..., g_T$ with d dimensional input x, a random forest g ensembles these trees by weightedly averaging their outputs, namely $g = \sum_{t=1}^{T} \lambda_t g_t$, where λ_t denotes the weight of tree t ($\sum_{t=1}^{T} \lambda_t = 1$).

As illustrated in Figure 2, a decision tree consists of nodes and a branch structure. The nodes are categorized into splits (triangle node), the nodes with two child nodes, and leaves (circle node), the nodes with no child node. At each split, we execute a binary query defined by a dimension index and a split point, i.e., in the form of $x_i \leq a$, where x_i denotes the ith dimension of the input x and $a \in \mathbb{R}$ is the split point. Starting from the root node, a sequence of queries leads the input down to a leaf node which corresponds to an output value.

To look for dominating points in a random forest, we follow the route in [86] that studies optimization over these models. Following the notations therein, we use $a_{i,j}$ to summarize the split point information from all trees in g, which

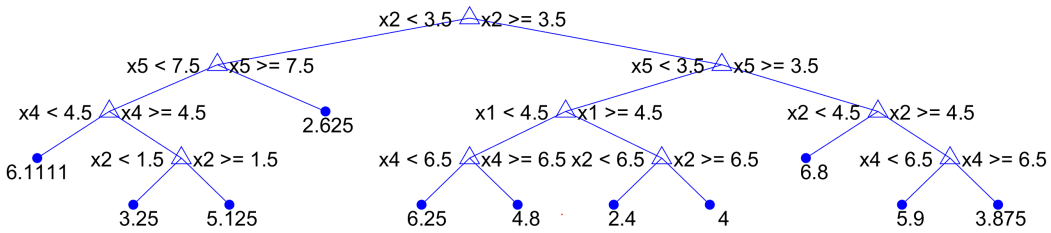

Fig. 2. An example of a decision tree.

denotes the jth unique split point for the ith dimension of the input x. Note that $a_{i,1} < a_{i,2} < \ldots < a_{i,K_i}$, where K_i is the number of unique split points for the ith dimension of x.

To represent the branch structure, we define $\mathbf{leaves}(t)$ as the set of leaves (terminal nodes) of tree t and $\mathbf{splits}(t)$ as the set of splits (non-terminal nodes) of tree t. In each split s, we let $\mathbf{left}(s)$ be the set of leaves that are accessible from the left branch (the query at s is true), and $\mathbf{right}(s)$ be the set of leaves that are accessible from the right branch (the query at s is false). For each node s, we use $\mathbf{V}(s) \in \{1, \ldots, d\}$ to denote the dimension that participates in the node and $\mathbf{C}(s) \in \{1, \ldots, K_{\mathbf{V}(s)}\}$ to denote the index of the split point on dimension i that participates in the query of s ($\mathbf{V}(s) = i$ and $\mathbf{C}(s) = j$ indicate the query $x_i \le a_{i,j}$). For each $l \in \mathbf{leaves}(t)$, $p_{t,l}$ denotes the output for the lth leaf in tree t.

To formulate the random forest optimization as an MIP, we introduce decision variables $z_{i,j}$ and $y_{t,l}$. Firstly, we use $z_{i,j}$ to locate the input x by linking its value to the split points $a_{i,j}$'s, where we have

$$z_{i,j} = I(x_i \le a_{i,j}),\ i = 1, \ldots, d,\ j = 1, \ldots, K_i. \tag{7}$$

In order to convert (7) into mixed integer constraints, we introduce an arbitrary large number $B \in \mathbb{R}^+$ which serves as the big-M coefficient [12] in our formulation. For any given problem, all dominating points must have finite coordinates. This implies that for large enough B we have $[-B, B]^d$ contain all dominating points. Thus, assuming we use a large enough B, we can let $x \in [-B, B]^d$ and $|a_{i,j}| \le B$. Then (7) is represented by the following constraints:

$$\begin{aligned} x_i &\le a_{i,j} + 2(1 - z_{i,j})B \\ x_i &> a_{i,j} - 2z_{i,j}B \\ z_{i,j} &= \{0, 1\}. \end{aligned}$$

Next we use $y_{t,l} = 1$ to denote that tree t outputs the prediction value $p_{t,l}$ on leaf l, and $y_{t,l} = 0$ otherwise. This allows us to represent the output of the random forest as

$$\sum_{t=1}^{T} \sum_{l \in \mathbf{leaves}(t)} \lambda_t p_{t,l} y_{t,l}$$

with $\sum_{l \in \mathbf{leaves}(t)} y_{t,l} = 1$. We use $\mathbf{z}, \mathbf{y}$ to represent the vectors of $z_{i,j}$ and $y_{t,l}$ respectively.

Lastly, we formulate the binary queries in a decision tree with these intermediate variables. This is achieved by forcing $y_{t,l}$ in the "unselected" branches to be 0. At each split s, we have

$$x_{\mathbf{V}(s)} > a_{\mathbf{V}(s),\mathbf{C}(s)} \Rightarrow \sum_{l \in \mathbf{left}(s)} y_{t,l} = 0$$
$$x_{\mathbf{V}(s)} \leq a_{\mathbf{V}(s),\mathbf{C}(s)} \Rightarrow \sum_{l \in \mathbf{right}(s)} y_{t,l} = 0,$$

which we reformulate with $\mathbf{z}$ into

$$\sum_{l \in \mathbf{left}(s)} y_{t,l} \leq z_{\mathbf{V}(s),\mathbf{C}(s)},\ \forall t \in \{1,...,T\},\ s \in \mathbf{splits}(t)$$
$$\sum_{l \in \mathbf{right}(s)} y_{t,l} \leq 1 - z_{\mathbf{V}(s),\mathbf{C}(s)},\ \forall t \in \{1,...,T\},\ s \in \mathbf{splits}(t).$$

Now we formulate (4) with $A = \emptyset$ as the following MIP

$$\min_{x,\mathbf{y},\mathbf{z}}\ (x-\mu)^T \Sigma^{-1} (x-\mu) \tag{8}$$
$$s.t.\ \sum_{t=1}^{T} \sum_{l \in \mathbf{leaves}(t)} \lambda_t p_{t,l} y_{t,l} \geq \gamma$$
$$\sum_{l \in \mathbf{leaves}(t)} y_{t,l} = 1,\ \forall t \in \{1,...,T\}$$
$$\sum_{l \in \mathbf{left}(s)} y_{t,l} \leq z_{\mathbf{V}(s),\mathbf{C}(s)},\ \forall t \in \{1,...,T\},\ s \in \mathbf{splits}(t)$$
$$\sum_{l \in \mathbf{right}(s)} y_{t,l} \leq 1 - z_{\mathbf{V}(s),\mathbf{C}(s)},\ \forall t \in \{1,...,T\},\ s \in \mathbf{splits}(t)$$
$$z_{i,j} \leq z_{i,j+1},\ \forall i \in \{1,...,d\},\ j \in \{1,...,K_i - 1\}$$
$$z_{i,j} \in \{0,1\},\ \forall i \in \{1,...,d\},\ j \in \{1,...,K_i\}$$
$$y_{t,l} \geq 0,\ \forall t \in \{1,...,T\},\ l \in \mathbf{leaves}(t)$$
$$x_i \leq a_{i,j} + 2(1 - z_{i,j})B,\ \forall i \in \{1,...,d\},\ j \in \{1,...,K_i\}$$
$$x_i > a_{i,j} - 2z_{i,j}B,\ \forall i \in \{1,...,d\},\ j \in \{1,...,K_i\}.$$

This formulation has a quadratic objective function and linear constraints. Similarly, we can formulate (4) with $A \neq \emptyset$ by adding linear constraints $(a_i - \mu)^T \Sigma^{-1} (x - a_i) < 0,\ \forall a_i \in A$ to (8). Note that both the number of decision variables and the number of constraints are linearly dependent on the total number of nodes in the random forest.

5.2 Tractable Formulation for Neural Network

A neural network $g(\cdot)$ is a network that connects a large number of computational units known as neurons [36, 59]. Depending on the task, this network bears a specific architecture that usually involves multiple layers of neurons and different operations over the neurons. For simplification, here we consider layers with consecutive architecture and each layer of the neural network only contains one specific structure.

The key part of the reformulation is to deal with the non-linearity brought by the maximum function. Our treatment of the maximum function follows from [111], which rewrites neural network structures into linear equations with binary variables.

In order to obtain tractable formulation for the constraint $g(x) \geq \gamma$, we independently handle each single layer in $g(\cdot)$. Assume we have l layers in $g(\cdot)$, where $g_i(\cdot)$ denotes the ith layer. Given input x, the output of the neural network can be represented as $g(x) = g_l(g_{l-1}(...g_1(x)))$. For convenience, we introduce x_i to denote the output of the ith layer (note that it is also the input for the $i+1$th layer). In other words, for the ith layer we have $x_i = g_i(x_{k-1})$. Using these notations, we can transform the constraint $g(x) \geq \gamma$ into a sequence of constraints:

$$
\begin{aligned}
& x_l \geq \gamma, \\
& x_l = g_l(x_{l-1}), \\
& x_{l-1} = g_{l-1}(x_{l-2}), \\
& \ldots, \\
& x_1 = g_1(x).
\end{aligned}
$$

This transformation makes clear that the constraints altogether are tractable if the constraint for each layer (i.e. $x_i = g_i(x_{i-1})$) is tractable . Note that both the number of decision variables and the number of constraints are linearly dependent on the total number of neurons in the neural network. In the rest of this section, we discuss the reformulation of neural network layers concerning different structures.

5.2.1 Fully Connected Layer. In a fully connected layer, each neuron performs a linear transformation on the input. We consider a layer with n neurons and the input for this layer is a vector $x \in \mathbb{R}^m$. We use $w_i \in \mathbb{R}^m$ and $b_i \in \mathbb{R}$ to denote the weight and bias respectively for the linear transformation in the ith neuron. Then the output of the ith neuron can be represented by $y_i = w_i^T x + b_i$. To summarize, the output of the layer, $y = [y_1; y_2; ...; y_n] \in \mathbb{R}^n$, is given by

$$y = W^T x + b,$$

where $W = [w_1, w_2, ..., w_n]$ and $b = [b_1; b_2; ...; b_n]$.

5.2.2 ReLU Layer. In a rectified linear unit (ReLU) layer, negative elements in the input are replaced by 0's. For the ith input, the output is given by $y_i = max\{x_i, 0\}$. This can be represented by

$$
\begin{aligned}
& y_i \leq x_i - l(1 - z_i), \\
& y_i \geq x_i, \\
& y_i \leq u z_i, \\
& y_i \geq 0, \\
& z_i \in \{0, 1\},
\end{aligned}
$$

where $z_i \in \{0, 1\}$ is a binary variable, u and l are the upper and lower bounds of the input respectively.

5.2.3 Normalization Layer. In a normalization layer, the input is normalized and linearly transformed to make the gradient descent algorithm more efficient. Again we assume the input is $x \in \mathbb{R}^m$ with a given normalization parameter $\mu \in \mathbb{R}^m$ and $\Sigma \in \mathbb{R}^{m \times m}$. Moreover, we have the transformation matrix $\gamma \in \mathbb{R}^{m \times m}$ and bias vector $\beta \in \mathbb{R}^m$. The output

is given by

$$y = \gamma\left(\Sigma^{-1/2}(x-\mu)\right) + \beta.$$

5.2.4 Pooling Layer. In a pooling layer, a "filter" that can be applied to adjacent elements in a vector or matrix goes through the input with a certain stride. Such type of layer is used to summarize "local" information and reduce the dimension of the input. Max pooling and average pooling are two types of commonly used filters.

Suppose the input is represented by matrix $x \in \mathbb{R}^{m_1 \times m_2}$, where x_{ij} denotes the element in the ith row jth column. The size of the filter is $s_1 \times s_2$ with stride (s_1, s_2). The output has size $y \in \mathbb{R}^{n_1, n_2}$, where $n_1 = m_1/s_1$ and $n_2 = m_2/s_2$. We assume that the value of s_1, s_2 are carefully chosen so that n_1 and n_2 are integers.

For average pooling layer, we have

$$y_{ij} = \frac{\sum_{r=(i-1)s_1+1}^{is_1} \sum_{c=(j-1)s_2+1}^{js_2} x_{rc}}{s_1 s_2}$$

for $i = 1, ..., n_1, j = 1, ..., n_2$.

For max pooling layer, we have $y_{ij} = \max_{(r,c)\in S_{ij}} x_{rc}$ for $i = 1, ..., n_1$, $j = 1, ..., n_2$, where $S_{ij} = \{(r, c) | r = (i-1)s_1 + 1, ..., is_1, c = (j-1)s_2 + 1, ..., js_2\}$. The tractable formulation is given by

$$\begin{aligned}
& y_{ij} \le x_{rc} - (u-l)(1-z_{rc}), && (r,c) \in S_{ij} \\
& y_{ij} \ge x_{rc}, && (r,c) \in S_{ij} \\
& \sum_{(r,c)\in S_{ij}} z_{rc} = 1 \\
& z_{rc} \in \{0,1\}, && (r,c) \in S_{ij}.
\end{aligned}$$

5.2.5 Convolutional Layer. In a convolutional layer, several filters are used to extract features from the input. The input of the layer is $x \in \mathbb{R}^{m_1, m_2}$. Suppose we have r filters and assume the filters have size $s_1 \times s_2$ with stride (t_1, t_2). We use $w_i \in \mathbb{R}^{t_1 t_2}$ and $b_i \in \mathbb{R}^{t_1 t_2}$ to denote the weight and bias for the ith filter. The output is $y \in \mathbb{R}^{n_1 \times n_2 \times r}$, where $n_1 = (m_1 - s_1)/t_1$ and $n_2 = (m_2 - s_2)/t_2$. Again we assume the numbers are carefully chosen so that n_1, n_2 are integers.

Then we have

$$\begin{aligned}
& y_{ijk} = w_k^T(\tilde{x}_{ij}) + b_k, \\
& \tilde{x}_{ij} = [x_{(i-1)t_1+1,(j-1)t_2+1}; x_{(i-1)t_1+2,(j-1)t_2+1}; ...; x_{(i-1)t_1+1,(j-1)t_2+2}, ...; x_{(i-1)t_1+s_1,(j-1)t_2+s_2}].
\end{aligned}$$

for integers $1 \le i \le n_1$, $1 \le j \le n_2$ and $1 \le k \le r$.

5.2.6 Reformulation in the Output Layer. Here we discuss the reformulation of the output layer, which also provides us clues on how other more general problems in classification tasks are potentially transformable into the constraint $g(x) \ge \gamma$. Although the output layer is usually highly nonlinear, we show how to formulate it as linear mixed-integer constraints.

In classification tasks, the neural network usually uses a softmax layer as the output layer for training purposes. Suppose the classification problem has n categories in total, the last layer inputs $x \in \mathbb{R}^n$ and outputs $y \in \mathbb{R}^n$ with $y_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$. The prediction for classification is determined by the maximum value of y_i. Indeed, the result is equivalent if we determine the categories by the maximum value of x_i.

When the constraint is $g(X) = i$ or $g(X) \neq i$, we can use this equivalence to reformulate the last layer (and therefore complete the formulation for the whole network). Specifically, $g(X) = i$ can be formulated as $x_i \geq x_j, for\ j \neq i$ and $g(X) \neq i$ can be formulated as $x_i \leq \max_{j \neq i} x_j$, where $j \neq i$ denotes j is an element for the set that contains all possible indexes except i. For tractable form, the latter formula can be further rewritten as:

$$
\begin{aligned}
& x_i \leq x_j + (1 - z_j)(u - l),\ j \neq i. \\
& \sum_{j \neq i} z_j \geq 1, \\
& z_j \in \{0, 1\},\ i \neq c.
\end{aligned}
$$

6 EXPERIMENTS

This section presents several experimental results using our Algorithm 1 for neural network and random forest predictors. In Section 6.1, we consider two simple toy examples. The first problem has one dominating point and the second problem has multiple dominating points. To illustrate the efficiency of the proposed IS scheme, we compare it with a naive IS scheme using uniform distribution. In Section 6.2, we consider a realistic problem generated from a classification data set with a high-dimensional feature space.

6.1 Toy Problems

We consider the rare-event set $\{x : g(x) \geq \gamma\}$ and the input X follows a Gaussian distribution $N(0, I\sigma^2)$, where I denotes the identity matrix and $\sigma^2 \in \mathbb{R}^+$. The prediction model g is trained with a data set with uniformly designed inputs, and labeled using a deterministic function denoted by y. In order to build a prediction model with reasonable quality, the inputs of the training data are generated from a bounded region $[l, u]^d$, where the region is chosen sufficiently large in terms of γ that setting $g(x)$ to $-\infty$ outside the region barely affects the target probability. As a result, whether we impose this bound or not does not affect the probability materially, and we choose to impose it since this setting provides a good and simple IS scheme (i.e., uniform distribution) for comparison.

Given the above setting, we consider a uniform IS scheme as a baseline method in our experiments. Consider a problem where X follows a distribution $f(x)$, and the set $\{x : g(x) \geq \gamma\}$ is known to lie inside $[l, u]^d$ where d is the dimension of the input variable X. The uniform IS estimator of $P(g(x) \geq \gamma)$ is given by

$$
Z_{uniform} = I(g(X) \geq \gamma) f(X)(u - l)^d,
$$

where X is generated from a uniform distribution on $[l, u]^d$. This estimator has a polynomially growing relative efficiency as the magnitude of the dominating points grows [71], but the efficiency also depends significantly on the size of the bounded set, i.e., l, u, d.

In the first example, we use the deterministic function

$$
y(x) = (x_1 - 5)^3 + (x_2 - 4.5)^3 + (x_1 - 1)^2 + x_2^2 + 500 \tag{9}
$$

to label the training samples. We generate 2,601 samples with input $x = [x_1, x_2]$ using a uniform grid over the space with a mesh of 0.1 on each coordinate over the bounded space $[0, 5]^2$. The dataset we obtained is denoted as $D = \{(X_n, Y_n)\}$. $g(x)$ is trained using D. We note that the region $[0, 5]^2$ is large enough in our experiments, so that $g(x)$ can be thought of as being set to $-\infty$ outside this box. For instance, when $\sigma^2 = 1$, the ratio of the probability of falling outside $[-5, 5]^d$

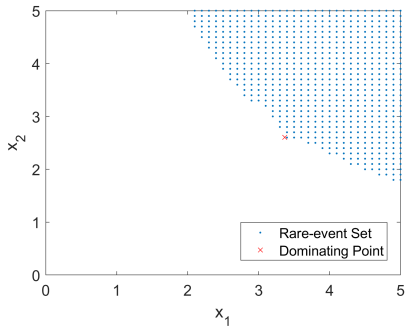

Fig. 3. Rare-event set and dominating points for the neural network (case 1).

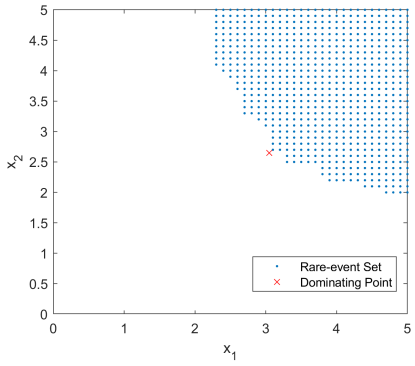

Fig. 4. Rare-event set and dominating points for the random forest (case 1).

(as $[0,5]^d$ is almost equivalent to $[-5,5]^d$ here) to the probability of interest (for the first example with $\gamma = 500$ or the second example with $\gamma = 8$) is smaller than 0.05, the largest ratio among all considered settings.

We first train a neural network predictor as $g(x)$. The neural network has 3 layers with 100 neurons in each of the 2 hidden layers, and all neurons are ReLU. To illustrate the rare-event set in the problem, we use $\gamma = 500$ in this example. The defined rare-event set is presented in Figure 3. We observe that the set is roughly convex and should have a single dominating point. We obtain the dominating point for the set at $(3.3676, 2.6051)$.

In our experiments, we first vary the value of γ to verify the asymptotic performance of the proposed IS estimator as γ increases. We then vary the value of σ^2 to create problems with different distribution setups, where a smaller σ^2 gives a rarer probability.

Figures 5 and 6 present the experimental results with fixed $\sigma^2 = 0.3$ and a varing γ based on 50,000 samples. Figure 5 shows that the proposed IS estimator provides similar estimates as the baseline estimator, while Figure 6 shows our estimator provides a better confidence interval width and the advantage grows slightly as γ increases.

In Figures 7 and 8, we present the experimental results for different variance values with $\gamma = 500$. Again we observe the proposed IS scheme provides smaller relative errors in all cases and the advantage increases with smaller variance (the relative error increases from 2.5 to 10 for the proposed IS and 5 to 55 for the uniform IS in the considered range of σ).

Next, we investigate how the size of the predictor would affect the efficiency of our proposed estimator. We note that a neural network with a larger size results in a larger number of linear pieces in the rare-event set formulation. To obtain rare-event sets with different numbers of linear pieces, we use neural networks with different number of neurons for training and subsequently building the rare-event sets. In particular, we vary the number of neurons in the second layer and keep other parameters fixed.

Table 1 presents the computation time for solving the mixed integer optimization under different cases. Although the numbers of constraints and variables increase by roughly 30% (from 150 total neurons to 200) as we increase the number of second layer neurons, there is no significant increase in the computation time. In Figures 9 and 10, we present the performances of our IS estimator. We observe that our IS estimator consistently outperforms the naive estimator as evidenced by the similar estimates in Figure 9 and the smaller relative errors in Figure 10.

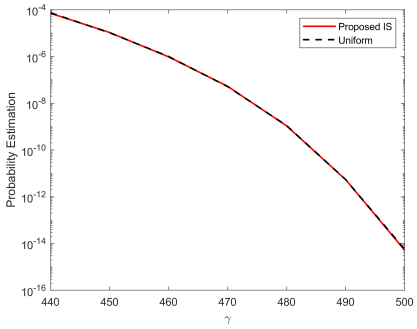

Fig. 5. Probability estimation with different γ. Neural network, case 1.

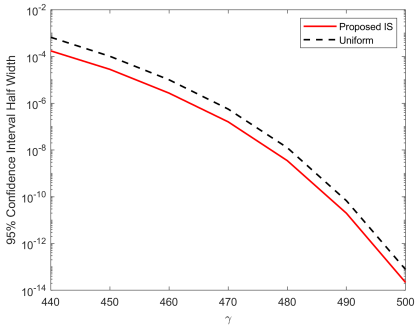

Fig. 6. 95% confidence interval half-width with different γ. Neural network, case 1.

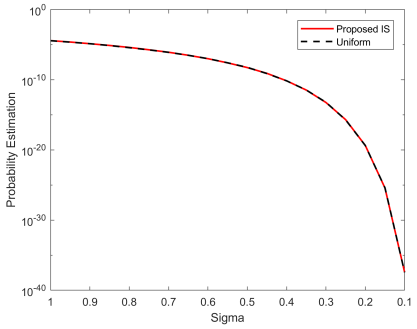

Fig. 7. Probability estimation with different distribution setups. Neural network, case 1.

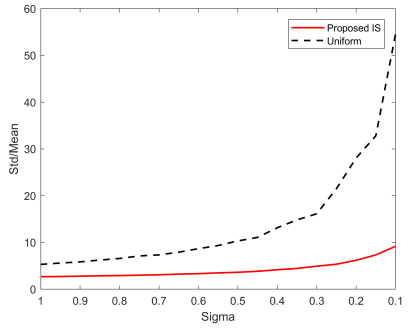

Fig. 8. Relative error with different distribution setups. Neural network, case 1.

Table 1. The computation time for solving the mixed integer optimization to obtain the first dominating point in the neural network defined rare-event set in case 1.

Number of Layer 2 Neurons	50	55	60	65	70	75	80	85	90	95	100
Number of Total Neurons	150	155	160	165	170	175	180	185	190	195	200
Computation Time (sec)	0.323	0.390	0.217	0.218	0.205	0.379	0.384	0.436	0.357	0.235	0.425

Next, we train a random forest $g(x)$, which ensembles three regression trees (see further training details in Appendix A). The three regression trees are averaged and each of them has around 600 nodes. Again we illustrate the rare-event set with $\gamma = 500$, which is presented in Figure 4. The dominating point is obtained by implementing Algorithm 1, which is located at $(3.05, 2.65)$.

Figures 11 and 12 show our results with random forest. In Figure 11, we observe that the estimates for the two IS schemes are similar in all considered cases. On the other hand, Figure 12 shows the relative error for the proposed IS is smaller in all considered σ. Moreover, as the rarity increases, the relative error of the proposed IS increases from roughly 2.5 to 5, whereas the relative error of the uniform IS increases from 5 to 40. The slower increasing rate indicates that the proposed IS scheme is more efficient and the outperformance is stronger for rarer problems.

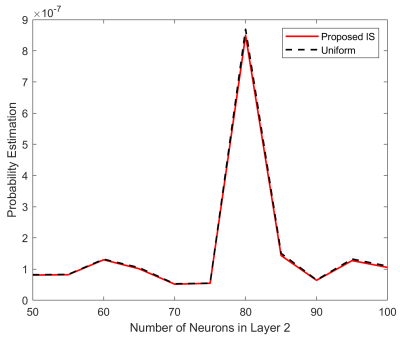

Fig. 9. Probability estimation with different different neural network sizes, case 1.

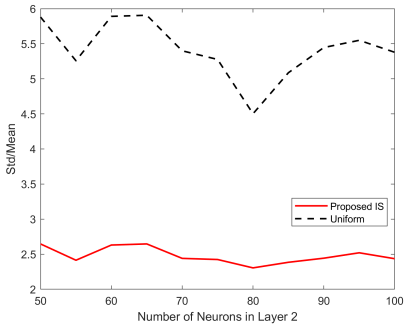

Fig. 10. Relative error with different neural network sizes, case 1.

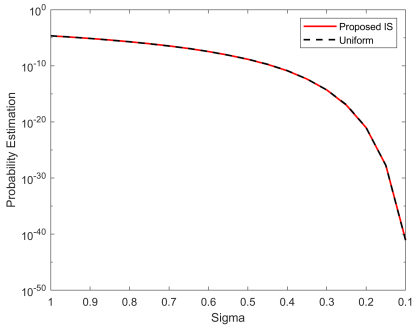

Fig. 11. Probability estimation with different distribution setups. Random forest, case 1.

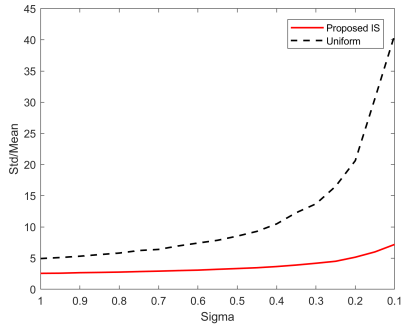

Fig. 12. Relative error with different distribution setups. Random forest, case 1.

We now consider true output values generated according to the function

$$y(x) = 10 \times e^{-\left(\frac{x_1-5}{3}\right)^2-\left(\frac{x_2-5}{4}\right)^2} + 10 \times e^{-{x_1}^2-(x_2-4.5)^2}. \tag{10}$$

Again we use a uniform grid over $[0, 5]^2$ with a mesh of 0.1 on each coordinate to train the predictors. The random forest ensembles three regression trees with around 600 nodes and the neural network with 2 hidden layers, 100 neurons in the first hidden layer and 50 neurons in the second hidden layer. All neurons in the neural network are ReLU.

For $\gamma = 8$, the shapes of the rare-event sets are shown in Figures 13 and 14. We observe that the set now consists of two disjoint regions and therefore we expect to obtain multiple dominating points. Using Algorithm 1, we obtain two dominating points in each case: $(0, 4.15)$ and $(3.75, 3.55)$ for the random forest model; $(0.113, 4.162)$ and $(4.187, 3.587)$ for the neural network model. Again we vary γ and σ^2 to obtain problems with different rarities and use 50,000 samples for each case.

Figures 15 and 16 shows the experiment results with fixed $\sigma^2 = 0.3$ and a varying γ. As in the first example, we observe that the IS estimator provides correct estimates with better confidence intervals through all considered cases. The experimental results with fixed $\gamma = 8$ varying σ^2 for the random forest predictor are shown in Figures 17 and 18,

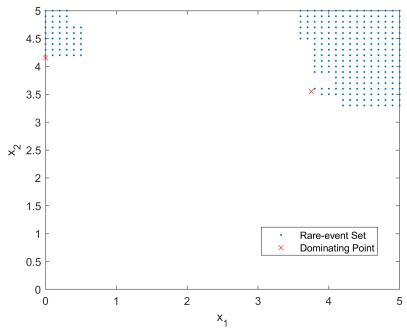

Fig. 13. Rare-event set and dominating point for the random forest (case 2).

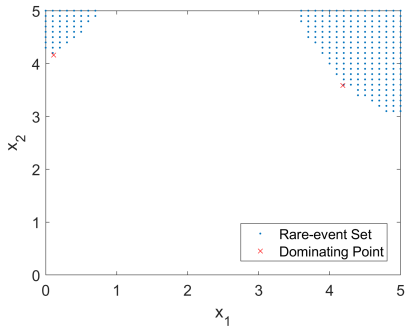

Fig. 14. Rare-event set and dominating point for the neural network (case 2).

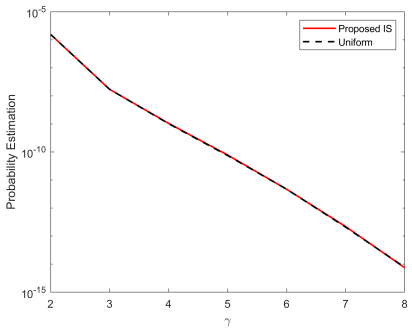

Fig. 15. Probability estimation with different γ. Neural network, case 2.

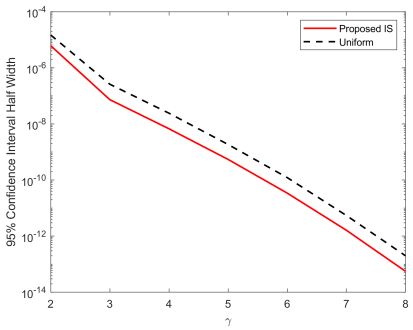

Fig. 16. 95% confidence interval half-width with different γ. Neural network, case 2.

and the results for the neural network predictor are shown in Figures 19 and 20. Similar to the previous problem, both IS schemes give similar estimates in all the cases, as observed in Figures 17 and 19. The relative errors shown in Figures 18 and 20 illustrate that, as the probability of interest decreases, the relative error ratio between the uniform IS and the proposed IS increases from 2 to around 5-6. We can conclude that the proposed IS scheme again outperforms the uniform IS and is more preferable as the rarity increases.

6.2 MAGIC Gamma Telescope Data Set

We study a rare-event probability estimation problem from a realistic classification task. The classification problem uses the MAGIC Gamma Telescope data set in the UCI Machine Learning Repository [10]. The problem is to classify images of electromagnetic showers collected by a ground-based atmospheric Cherenkov gamma telescope. The features of the data are 10-dimensional characteristic parameters of the images and the data set contains 19020 data points in total. We provide some descriptive statistics of the data set in Table 2. Studies [23, 47, 106] use machine learning predictors to discriminate images caused by a "signal" (primary gammas) from those initiated by the "background" (cosmic rays in the upper atmosphere).

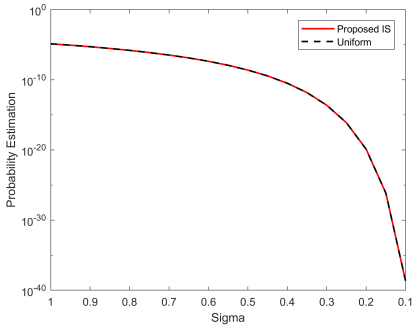

Fig. 17. Probability estimation with different distribution setups. Random forest, case 2.

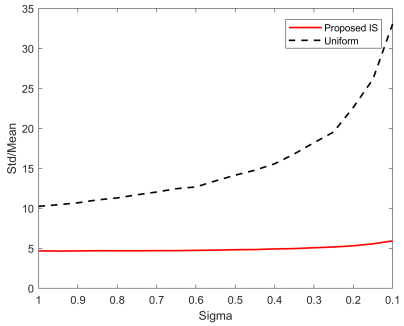

Fig. 18. Relative error with different distribution setups. Random forest, case 2.

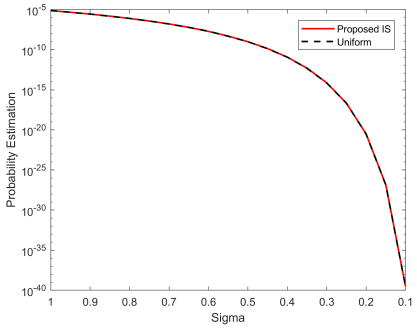

Fig. 19. Probability estimation with different distribution setups. Neural network, case 2.

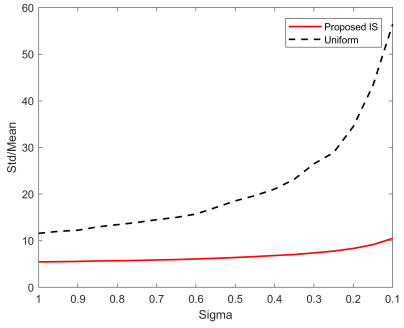

Fig. 20. Relative error with different distribution setups. Neural network, case 2.

Table 2. Descriptive statistics of the MAGIC Gamma Telescope Data Set. "Std" denotes the standard deviation and "CoV" denotes the coefficient of variation (ratio of the standard deviation to the mean).

Coefficient Index	1	2	3	4	5	6	7	8	9	10
Mean	53.250	22.181	2.825	0.380	0.215	-4.332	10.546	0.250	27.646	193.818
Std	42.365	18.346	0.473	0.183	0.111	59.206	51.000	20.827	26.104	74.732
CoV	0.796	0.827	0.167	0.481	0.515	-13.668	4.836	83.401	0.944	0.386
Min	4.284	0.000	1.941	0.013	0.000	-457.916	-331.780	-205.895	0.000	1.283
Max	334.177	256.382	5.323	0.893	0.675	575.241	238.321	179.851	90.000	495.561
Median	37.148	17.140	2.740	0.354	0.197	4.013	15.314	0.666	17.680	191.851

To train the predictors, we allocate 15,000 data points as the training set and use the remaining 4,020 data points as the testing set. All data were normalized to avoid scaling issues in training. We train a random forest that ensembles 10 random trees to achieve 85.6% testing set accuracy. For neural network, we use 2 hidden layers with 20 neurons and achieved 87% testing set accuracy.

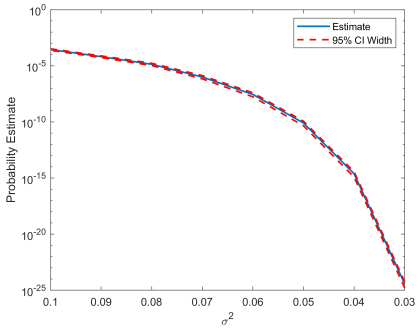

Fig. 21. Probability estimation with different distribution setups. Random forest, MAGIC.

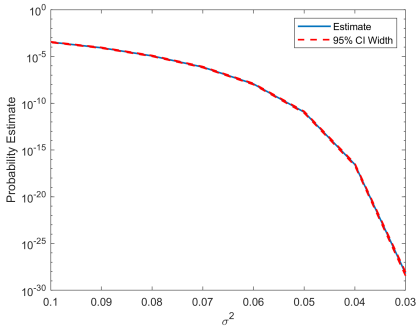

Fig. 22. Probability estimation with different distribution setups. Neural network, MAGIC.

The rare-event probability of interest is the statistical robustness metric (Example 3.1) of the two trained predictors. Specifically, we consider a testing data point, say with input x and true label y, that is correctly predicted in both predictors (the predicted value $g(x)$ is consistent with y). Then we perturb the input x with a Gaussian noise $\epsilon \sim N(0, I\sigma^2)$ and estimate the probability of $P(g(x+\epsilon) \neq y)$, where we use uniform variance for each dimension because the input space was normalized. In our experiment, we vary the value of σ^2 to construct rare-event with different rarities. Note that, as discussed in Example 3.1, $P(g(x+\epsilon) \neq y)$ can be transformed into the format considered in this paper, i.e. $P(g(X) > \gamma)$.

First, we implement Algorithm 1 to obtain dominating points for the rare-event sets $\{g(x+\epsilon) \neq y\}$ with random forest and neural network as $g(\cdot)$ respectively. We obtain 53 dominating points for the rare-event sets associated with the random forest predictor and 217 dominating points in the neural network case. The IS distributions are constructed using these dominating points. In both problems, σ^2 ranges from 0.03 to 0.1 and we use 50,000 samples to estimate each target rare-event probabilities.

The experimental results for the random forest and neural network are presented in Figures 21 and 22 respectively. We observe that the estimates are very accurate in all experiments (with different rarities), which are indicated by the tight 95% confidence intervals. These results show that our proposed IS scheme performs well with large numbers of dominating points and in relatively high-dimensional problems.

7 PROOFS OF THEOREMS

Throughout this section, we write $f_1(\gamma) \sim f_2(\gamma)$ if $\lim_{\gamma\to\infty} f_1(\gamma)/f_2(\gamma) = 1$ and write $f_1(\gamma) \overset{poly}{\sim} f_2(\gamma)$ if $f_1(\gamma)/f_2(\gamma)$ changes at most polynomially in γ. Unless otherwise defined, we use x_i to denote the i-th component of a vector x. For any vectors $x, y \in \mathbb{R}^d$, we write $x \geq y$ if $x_i \geq y_i$ for any $i = 1, \ldots, d$. For any index sets $I, J \subset \{1, \ldots, d\}$ and any $x \in \mathbb{R}^d$, $A \in \mathbb{R}^{d\times d}$, we use x_I to denote the subvector $(x_i)_{i\in I}$ and use A_{IJ} to denote the submatrix $(A_{ij})_{i\in I, j\in J}$.

First of all, we adapt Theorem 4.1 in [64] to obtain the following lemma.

Lemma 7.1. *Let Y be a d-dimensional Gaussian random vector with zero mean and positive definite covariance matrix $\tilde{\Sigma}$. Suppose that $\tilde{s} = \tilde{s}(\gamma) \notin [-\infty, 0]^d$ is a vector in $[-\infty, \infty)^d$ such that as $\gamma \to \infty$, at least one of its components goes to ∞. Use y^* to denote $\arg\min_{y\geq\tilde{s}} y^T\tilde{\Sigma}^{-1}y$. Then by Proposition 2.1 in [64], we know that there exists a unique set $I \subset \{1, \cdots, d\}$*

such that

$$1 \le |I| \le d; \tag{11a}$$

$$y_I^* = \tilde{s}_I \neq \mathbf{0}_I; \tag{11b}$$

$$\text{If } J := \{1, \dots, d\} \setminus I \neq \emptyset, \text{ then } y_J^* = -(\tilde{\Sigma}^{-1})_{JJ}^{-1}(\tilde{\Sigma}^{-1})_{JI}\tilde{s}_I \ge \tilde{s}_J; \tag{11c}$$

$$\forall i \in I, e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I > 0; \tag{11d}$$

$$\min_{y \ge \tilde{s}} y^T\tilde{\Sigma}^{-1}y = (y^*)^T\tilde{\Sigma}^{-1}y^* > 0. \tag{11e}$$

We suppose that for sufficiently large γ, the set I does not change with γ and if $J \neq \emptyset$, $\lim_{\gamma\to\infty}(\tilde{s} - y^*)_J = \tilde{s}_J^*$ where $\tilde{s}^*$ is a constant vector in $[-\infty, \infty)^{|J|}$. Suppose further that $\forall i \in I$, $e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I$ either goes to ∞ or is a positive constant. Then as $\gamma \to \infty$, we have that

$$P(Y \ge \tilde{s}) \sim C\frac{\exp\{-(y^*)^T\tilde{\Sigma}^{-1}y^*/2\}}{\prod_{i\in I} e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I}$$

where C is a positive constant in γ.

Before showing the proof, we provide a brief and intuitive explanation on the index set I as defined in the lemma. We minimize $y^T\tilde{\Sigma}^{-1}y$ subject to $y \ge \tilde{s}$. Among the constraints, $y_I \ge \tilde{s}_I$ is crucial while $y_J \ge \tilde{s}_J$ could be removed without affecting the optimal solution. Thus, the original optimization problem is equivalent to minimizing $y^T\tilde{\Sigma}^{-1}y$ subject to $y_I = \tilde{s}_I, y_J \in \mathbb{R}^{|J|}$. For example, if $d = 2, \tilde{s} = (1, 0)^T$ and $\tilde{\Sigma}$ is the identity matrix, then $I = \{1\}$ and $J = \{2\}$ since $y_1 \ge 1$ could not be removed while $y_2 \ge 0$ could. Now we prove the lemma:

Proof of Lemma 7.1. Given $x \in \mathbb{R}^d$, we define the transformation $\tilde{x}$ in the following way: $\tilde{x}_i = (e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I)^{-1}x_i, \forall i \in I; \tilde{x}_J = x_J$. Using (3.4) in [64], we know that

$$(x + y^*)^T\tilde{\Sigma}^{-1}(x + y^*) = x^T\tilde{\Sigma}^{-1}x + 2x_I^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I + (y^*)^T\tilde{\Sigma}^{-1}y^*,$$

and thus

$$\begin{aligned}\phi(\tilde{x} + y^*; 0, \tilde{\Sigma}) &= (2\pi)^{-\frac{d}{2}}|\tilde{\Sigma}|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\left[\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x} + 2\tilde{x}_I^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I + (y^*)^T\tilde{\Sigma}^{-1}y^*\right]\right\}\\ &= (2\pi)^{-\frac{d}{2}}|\tilde{\Sigma}|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}\left[\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x} + 2x_I^T\mathbf{1}_I + (y^*)^T\tilde{\Sigma}^{-1}y^*\right]\right\}\end{aligned}$$

Then we get that

$$\begin{aligned}&P(Y \ge \tilde{s})\\ =&\int_{y\ge\tilde{s}} \phi(y; 0, \tilde{\Sigma})\mathrm{d}y\\ =&\int_{\tilde{x}\ge\tilde{s}-y^*} \phi(\tilde{x} + y^*; 0, \tilde{\Sigma})\mathrm{d}\tilde{x}\\ =&\left(\prod_{i\in I} e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I\right)^{-1}\int_{x\ge\tilde{s}-y^*} \phi(\tilde{x} + y^*; 0, \tilde{\Sigma})\mathrm{d}x \quad \text{(In the integrand, } \tilde{x} \text{ can be viewed as a function of } x.)\\ =&(2\pi)^{-\frac{d}{2}}|\tilde{\Sigma}|^{-\frac{1}{2}}\left(\prod_{i\in I} e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I\right)^{-1}\exp\{-(y^*)^T\tilde{\Sigma}^{-1}y^*/2\}\int_{x\ge\tilde{s}-y^*}\exp\{-\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}/2 - x_I^T\mathbf{1}_I\}\mathrm{d}x.\end{aligned}$$

Apparent from the above, it suffices to show that $\int_{x\geq\tilde{s}-y^*}\exp\{-\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}/2-x_I^T\mathbf{1}_I\}\mathrm{d}x$ converges to a positive constant as $\gamma\to\infty$. We will prove this result via applying the dominated convergence theorem. We first need to derive an integrable upper bound for the integrand. Indeed, using (3.6) in [64] we know that

$$\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}+2\tilde{x}_I^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I+(y^*)^T\tilde{\Sigma}^{-1}y^*\geq\tilde{x}_J^T(\tilde{\Sigma}_{JJ})^{-1}\tilde{x}_J+2\tilde{x}_I^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I+(y^*)^T\tilde{\Sigma}^{-1}y^*$$

and hence $\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}\geq\tilde{x}_J^T(\tilde{\Sigma}_{JJ})^{-1}\tilde{x}_J=x_J^T(\tilde{\Sigma}_{JJ})^{-1}x_J$. Thus

$$\exp\{-\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}/2-x_I^T\mathbf{1}_I\}\leq\exp\{-x_J^T(\tilde{\Sigma}_{JJ})^{-1}x_J/2-x_I^T\mathbf{1}_I\}.$$

Moreover, we have that

$$\begin{aligned}\int_{x\geq\tilde{s}-y^*}\exp\{-x_J^T(\tilde{\Sigma}_{JJ})^{-1}x_J/2-x_I^T\mathbf{1}_I\}\mathrm{d}x&\leq\int_{x_I\geq\mathbf{0}_I,x_J\in\mathbb{R}^{|J|}}\exp\{-x_J^T(\tilde{\Sigma}_{JJ})^{-1}x_J/2-x_I^T\mathbf{1}_I\}\mathrm{d}x\\&=\int_{\mathbb{R}^{|J|}}\exp\{-x_J^T(\tilde{\Sigma}_{JJ})^{-1}x_J/2\}\mathrm{d}x_J<\infty.\end{aligned}$$

To investigate the limit of $\exp\{-\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}/2-x_I^T\mathbf{1}_I\}$, we further partition I into $I_1=\{i\in I:e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I\to\infty\}$ and $I_2=\{i\in I:e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I$ is a positive constant$\}$. By the definition, we know that for any given $x\in\mathbb{R}^d$, $\tilde{x}_i\to 0$ for $i\in I_1$ and $\tilde{x}_i$ is a constant for $i\in I_2$ or J. Then we get that for any x,

$$\begin{aligned}\lim_{\gamma\to\infty}\exp\{-\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}/2-x_I^T\mathbf{1}_I\}&=\exp\left\{-\frac{1}{2}\left[\tilde{x}_{I_2}^T(\tilde{\Sigma}^{-1})_{I_2I_2}\tilde{x}_{I_2}+2\tilde{x}_{I_2}^T(\tilde{\Sigma}^{-1})_{I_2J}\tilde{x}_J+\tilde{x}_J^T(\tilde{\Sigma}^{-1})_{JJ}\tilde{x}_J\right]-x_I^T\mathbf{1}_I\right\}\\&=\exp\left\{-\frac{1}{2}\left[\tilde{x}_{I_2}^T(\tilde{\Sigma}^{-1})_{I_2I_2}\tilde{x}_{I_2}+2\tilde{x}_{I_2}^T(\tilde{\Sigma}^{-1})_{I_2J}x_J+x_J^T(\tilde{\Sigma}^{-1})_{JJ}x_J\right]-x_I^T\mathbf{1}_I\right\}.\end{aligned}$$

By applying the dominated convergence theorem, we get that

$$\begin{aligned}&\lim_{\gamma\to\infty}\int_{x\geq\tilde{s}-y^*}\exp\{-\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}/2-x_I^T\mathbf{1}_I\}\mathrm{d}x\\&=\int\int_{x_I\geq\mathbf{0}_I,x_J\geq\tilde{s}_J^*}\exp\left\{-\frac{1}{2}\left[\tilde{x}_{I_2}^T(\tilde{\Sigma}^{-1})_{I_2I_2}\tilde{x}_{I_2}+2\tilde{x}_{I_2}^T(\tilde{\Sigma}^{-1})_{I_2J}x_J+x_J^T(\tilde{\Sigma}^{-1})_{JJ}x_J\right]-x_I^T\mathbf{1}_I\right\}\mathrm{d}x_I\mathrm{d}x_J\\&=\int\int_{x_{I_2}\geq\mathbf{0}_{I_2},x_J\geq\tilde{s}_J^*}\exp\left\{-\frac{1}{2}\left[\tilde{x}_{I_2}^T(\tilde{\Sigma}^{-1})_{I_2I_2}\tilde{x}_{I_2}+2\tilde{x}_{I_2}^T(\tilde{\Sigma}^{-1})_{I_2J}x_J+x_J^T(\tilde{\Sigma}^{-1})_{JJ}x_J\right]-x_{I_2}^T\mathbf{1}_{I_2}\right\}\mathrm{d}x_{I_2}\mathrm{d}x_J.\end{aligned}$$

This shows that $\int_{x\geq\tilde{s}-y^*}\exp\{-\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}/2-x_I^T\mathbf{1}_I\}\mathrm{d}x$ converges to a positive constant as $\gamma\to\infty$, and hence we have proved the theorem. □

Now we apply Lemma 7.1 and the techniques in its proof to derive the following result:

Lemma 7.2. *Suppose that $X\sim N(\mu,\Sigma)$ where $\mu\in\mathbb{R}^d$ and $\Sigma\in\mathbb{R}^{d\times d}$ is positive definite. Let $A\sim\mathbb{R}^{m\times d}$ be a constant matrix and $t\in\mathbb{R}^m$ be a vector. In particular, $t_1=\gamma+c$ for some constant $c\in\mathbb{R}$ and $t_2,\ldots,t_m$ are all constants in $\mathbb{R}$. Assume that $P(AX\geq t)>0$ for any $\gamma\in\mathbb{R}$. Define $x^*=\arg\min\{(x-\mu)^T\Sigma^{-1}(x-\mu):Ax\geq t\}$. Then*

(i) Use A_i to denote the i-th row vector of A and define $\mathcal{A}(x)=\{1\leq i\leq m:A_i^Tx=t_i\}$ for $x\in\mathbb{R}^d$. For sufficiently large γ, $\mathcal{A}(x^*)$ does not change with γ.

(ii) For sufficiently large γ, each component of x^* is affine in γ.

(iii) As $\gamma\to\infty$,

$$P(AX\geq t)\overset{poly}{\sim}\exp\{-(x^*-\mu)^T\Sigma^{-1}(x^*-\mu)/2\}.$$

Proof of Lemma 7.2. For simplicity, we denote the polyhedron $\{x \in \mathbb{R}^d : Ax \geq t\}$ as P_1.

(i&ii) Note that x^* is the optimal solution to a quadratic programming problem. It is known that

$$x^* = \arg\min\{(x-\mu)^T\Sigma^{-1}(x-\mu) : A_i^T x = t_i, \forall i \in \mathcal{A}(x^*)\}. \tag{12}$$

Moreover, as γ grows, actually only the first constraint $A_1^T x \geq t_1 = \gamma + c$ shifts with γ while the other $m-1$ constraints keep unchanged. Thus we must have $1 \in \mathcal{A}(x^*)$ for sufficiently large γ. Indeed, if $1 \notin \mathcal{A}(x^*)$, then from (12), x^* must belong to $\{\arg\min\{(x-\mu)^T\Sigma^{-1}(x-\mu) : A_i^T x = t_i, \forall i \in \bar{I}\} : \bar{I} \subset \{2,\dots,m\}\}$, which is a finite set of constant vectors. However, we have that $A_1^T x^* \geq t_1 = \gamma + c$, so when γ is large enough, x^* cannot be one of these constant vectors and hence $1 \in \mathcal{A}(x^*)$.

We consider the "candidate points" defined as follows. For any fixed index set $\tilde{I} \subset \{1,\dots,m\}$ such that $1 \in \tilde{I}$, $\{x \in \mathbb{R}^d : A_i^T x = t_i, i \in \tilde{I}\} \neq \emptyset$ and the constraints $A_i^T x = t_i, i \in \tilde{I}$ are linearly independent for sufficiently large γ (we call such $\tilde{I}$ as valid), we solve $x^*(\tilde{I}) = \arg\min\{(x-\mu)^T\Sigma^{-1}(x-\mu) : A_i^T x = t_i, \forall i \in \tilde{I}\}$. If $x^*(\tilde{I})$ is feasible for the original problem, i.e. $x^*(\tilde{I}) \in P_1$, then we call $x^*(\tilde{I})$ a candidate point.

We note that the total number of valid $\tilde{I}$ is finite since $\tilde{I}$ is always a subset of $\{1,\dots,m\}$. Without loss of generality, from now on we assume that γ is large enough such that $1 \in \mathcal{A}(x^*)$ and for any valid $\tilde{I}$, $\{x \in \mathbb{R}^d : A_i^T x = t_i, i \in \tilde{I}\} \neq \emptyset$ and $A_i^T x = t_i, i \in \tilde{I}$ are linearly independent. In this case, x^* is the candidate point which attains the minimum objective value.

First, we show that for any valid $\tilde{I}$, each component of $x^*(\tilde{I})$ is affine in γ. Suppose that $\tilde{I} = \{i_1,\dots,i_{|\tilde{I}|}\}$ with $i_1 < \cdots < i_{|\tilde{I}|}$. We have that $A_{i_j}, j = 1,\dots,|\tilde{I}|$ are linearly independent. Let $A(\tilde{I}) \in \mathbb{R}^{d\times d}$ be a constant invertible matrix whose j-th row vector is A_{i_j} for $j = 1,\dots,|\tilde{I}|$. Consider the transformation $y = A(\tilde{I})(x-\mu)$ and solve $y^*(\tilde{I}) = \arg\min\{y^T(A(\tilde{I})^{-1})^T\Sigma^{-1}A(\tilde{I})^{-1}y : y_j = t_{i_j} - A_{i_j}^T\mu, j = 1,\dots,|\tilde{I}|\}$. We have that $x^*(\tilde{I}) = A(\tilde{I})^{-1}y^*(\tilde{I}) + \mu$. To ease the notation, we denote $\Sigma' = A(\tilde{I})^T\Sigma A(\tilde{I})$, $t' = t_{\tilde{I}} - (A\mu)_{\tilde{I}}$, $I' = \{1,\dots,|\tilde{I}|\}$ and $J' = \{1,\dots,d\} \setminus I'$. Then

$$\begin{aligned} y^*(\tilde{I}) &= \arg\min\{y^T\Sigma'^{-1}y : y_{I'} = t'\} \\ &= \arg\min\{y_{I'}^T(\Sigma'^{-1})_{I'I'}y_{I'} + 2y_{I'}^T(\Sigma'^{-1})_{I'J'}y_{J'} + y_{J'}^T(\Sigma'^{-1})_{J'J'}y_{J'} : y_{I'} = t'\}. \end{aligned}$$

By solving the above problem, we get that $y^*(\tilde{I})_{I'} = t'$ and $y^*(\tilde{I})_{J'} = -(\Sigma'^{-1})_{J'J'}^{-1}(\Sigma'^{-1})_{J'I'}t'$. By the definition, for fixed index set $\tilde{I}$, Σ' is a constant matrix. Besides, $t_1' = t_1 - (A\mu)_1$ is an affine function in γ while other components of t' are all constants. Hence, each component of $y^*(\tilde{I})$ is affine in γ. As a result, each component of $x^*(\tilde{I})$ is also affine in γ.

To check whether $x^*(\tilde{I})$ is feasible, it is equivalent to check whether $A_i^T x^*(\tilde{I}) \geq t_i$ for any $i \notin \tilde{I}$. We know that for $i \notin \tilde{I}$ (which implies that $i \neq 1$), $A_i^T x^*(\tilde{I})$ is affine in γ while t_i is a constant. Hence, for sufficiently large γ, it is determined whether $x^*(\tilde{I})$ is a candidate point or not. Again, the total number of valid $\tilde{I}$ is finite. Therefore, $\{\tilde{I} : x^*(\tilde{I})$ is a candidate point$\}$ does not change for large γ.

Finally, for each $\tilde{I}$ such that $x^*(\tilde{I})$ is a candidate point for sufficiently large γ, we have that the objective value $(x^*(\tilde{I})-\mu)^T\Sigma^{-1}(x^*(\tilde{I})-\mu)$ is a quadratic function of γ. Recall that x^* is the candidate point with minimum objective value. Thus, when γ is sufficiently large, $\{\tilde{I} : x^* = x^*(\tilde{I})\}$ must be non-empty and fixed. We pick a specific $\tilde{I}$ such that $x^* = x^*(\tilde{I})$ for sufficiently large γ. Since we have proved that each component of $x^*(\tilde{I})$ is affine in γ, we get statement (ii). Then when γ is large enough, for each i, it is determined whether $A_i^T x^* = t_i$, i.e. $i \in \mathcal{A}(x^*)$, which completes the proof of (i).

(iii) In this proof, we will construct a superset and a subset of P_1, and hence develop an upper bound and a lower bound for $P(X \in P_1)$. Then it suffices to show that both bounds are approximately $\exp\{-(x^*-\mu)^T\Sigma^{-1}(x^*-\mu)/2\}$ up to polynomial factors.

First, we construct the superset of P_1 by removing constraints. Following the above proof, we can find a maximal valid index set $\tilde{I} = \{i_1, \ldots, i_{m'}\}$ such that $1 = i_1 < \cdots < i_{m'}$ and $x^* = x^*(\tilde{I})$ for sufficiently large γ. Intuitively, $\{A_{i_j}^T x \geq t_{i_j}, j = 1, \ldots, m'\}$ is the maximal linearly independent subset of active constraints at x^*. If $m' < d$, then we can add redundant constraints in the form of $x_{k_l} \geq -\infty, l = 1, \cdots, d - m'$ such that we get d linearly independent constraints now. More specifically, let

$$B = \begin{pmatrix} A_{i_1}^T \\ \vdots \\ A_{i_{m'}}^T \\ e_{k_1}^T \\ \vdots \\ e_{k_{d-m'}}^T \end{pmatrix}, s = \begin{pmatrix} t_{i_1} \\ \vdots \\ t_{i_{m'}} \\ -\infty \\ \vdots \\ -\infty \end{pmatrix}.$$

By the construction, we get that B is a $d \times d$ constant invertible matrix. Denote $P_2 = \{x \in \mathbb{R}^d : Bx \geq s\}$. Since P_2 is obtained by removing constraints from P_1, we have that $P_1 \subset P_2$ and thus $P(X \in P_1) \leq P(X \in P_2)$. Now we develop the asymptotic result of $P(X \in P_2)$, where we directly apply Lemma 7.1.

We know that $Y := B(X - \mu) \sim N(0, \tilde{\Sigma})$ where $\tilde{\Sigma} = B\Sigma B^T$ is positive definite. We denote $y^* = \arg\min\{y^T\tilde{\Sigma}^{-1}y : y \geq \tilde{s}\}$ where $\tilde{s} = s - B\mu$. It is easy to verify that $y^* = B(x^* - \mu)$ and $(y^*)^T\tilde{\Sigma}^{-1}y^* = (x^* - \mu)^T\Sigma^{-1}(x^* - \mu)$. From (ii), we know that each component of y^* is also affine in γ for large γ.

Now we verify the assumptions of Lemma 7.1. Recall that under our settings, $s_1 = \gamma + c$ for some constant $c \in \mathbb{R}$ so $\tilde{s}_1 \to \infty$ as $\gamma \to \infty$. We still use the symbol I to denote the set that satisfies (11). By the definition, clearly $\{m'+1, \ldots, d\} \subset J = \{1, \ldots, d\} \setminus I$. Basically, I is the minimal subset of $\{1, \ldots, m'\}$ such that $x^* = x^*(\{i_j : j \in I\})$. Following the previous proof, we know that $1 \in I$ and I does not change for sufficiently large γ. Moreover, we know that $y^* \geq \tilde{s}$ and each component of y^* is affine in γ, then the limit $\lim_{\gamma\to\infty}(\tilde{s} - y^*)_J$ exists in $[-\infty, 0]^{|J|}$. Indeed, for $j \in J \cap \{2, \ldots, m'\}$, $\tilde{s}_j$ is a constant and then $\tilde{s}_j - y_j^*$ converges to $-\infty$ or a nonpositive constant while for $j \in \{m'+1, \ldots, d\}$, $\tilde{s}_j - y_j^* \equiv -\infty$. Finally, for any $i \in I$, we know that $e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I > 0$ and it is an affine function of γ, and thus either it goes to ∞ or it is a positive constant as $\gamma \to \infty$.

In conclusion, all the assumptions of Lemma 7.1 hold in this case. Therefore, we get that

$$P(X \in P_2) \sim C\frac{\exp\{-(y^*)^T\tilde{\Sigma}^{-1}y^*/2\}}{\prod_{i\in I} e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I} = C\frac{\exp\{-(x^*-\mu)^T\Sigma^{-1}(x^*-\mu)/2\}}{\prod_{i\in I} e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I}$$

for some positive constant C, which implies that

$$P(X \in P_2) \overset{poly}{\sim} \exp\{-(x^*-\mu)^T\Sigma^{-1}(x^*-\mu)/2\}.$$

Clearly, if $P_1 = P_2$, then we have proved statement (iii). From now on, we assume that $P_1 \neq P_2$. In this case, P_2 is a relaxation of P_1 by removing inactive constraints at x^*. That is, for any $x \in \overline{P_2 \setminus P_1}$, there exists $i \in \{1, \ldots, m\}$ such that $A_i^T x \leq t_i$ while $A_i^T x^* > t_i$. In particular, $x^* \notin \overline{P_2 \setminus P_1}$.

Next, we construct the subset of P_1 by selecting a small neighborhood around x^*. Denote $x^{**} = \arg\min\{(x - x^*)^T\Sigma^{-1}(x - x^*) : x \in \overline{P_2 \setminus P_1}\}$. Note that $\overline{P_2 \setminus P_1}$ can be expressed as the union of finite polyhedrons, each of which is formed by a shifting constraint and some constant constraints like P_1. Similar to the previous arguments, we could derive that $(x^{**} - x^*)^T\Sigma^{-1}(x^{**} - x^*) \geq 0$ is a quadratic function of γ for large γ. Thus we know that $(x^{**} - x^*)^T\Sigma^{-1}(x^{**} - x^*)$ either goes to ∞ or stays a nonnegative constant as $\gamma \to \infty$. However, if $(x^{**} - x^*)^T\Sigma^{-1}(x^{**} - x^*) = 0$ for sufficiently large γ, then we have that $x^{**} = x^*$, which contradicts with $x^* \notin \overline{P_2 \setminus P_1}$. Therefore, there exists a constant $0 < \varepsilon < 1$ such that for sufficiently large γ, $(x^{**} - x^*)^T\Sigma^{-1}(x^{**} - x^*) > \varepsilon^2$, and hence $\overline{P_2 \setminus P_1} \subset \{x \in \mathbb{R}^d : (x - x^*)^T\Sigma^{-1}(x - x^*) > \varepsilon^2\}$. Thus, $\{x \in \mathbb{R}^d : (x - x^*)^T\Sigma^{-1}(x - x^*) \leq \varepsilon^2\} \cap P_1 = \{x \in \mathbb{R}^d : (x - x^*)^T\Sigma^{-1}(x - x^*) \leq \varepsilon^2\} \cap P_2$ for sufficiently large γ. Correspondingly, there exists $\varepsilon' > 0$ such that $\{x \in \mathbb{R}^d : \|x\|_\infty \leq \varepsilon'\} \subseteq \{x \in \mathbb{R}^d : x^T\Sigma^{-1}x \leq \varepsilon^2\}$.

Still we define $Y = B(X - \mu) \sim N(0, \tilde{\Sigma})$. Then we get that

$$\begin{aligned} P(X \in P_1) &\geq P((X - x^*)^T\Sigma^{-1}(X - x^*) \leq \varepsilon^2, X \in P_1) \\ &= P((X - x^*)^T\Sigma^{-1}(X - x^*) \leq \varepsilon^2, X \in P_2) \\ &= P((Y + B\mu - Bx^*)^T\tilde{\Sigma}^{-1}(Y + B\mu - Bx^*) \leq \varepsilon^2, Y \geq \tilde{s}). \end{aligned}$$

Similar to the proof of Lemma 7.1, we have that

$$\begin{aligned} &P((Y + B\mu - Bx^*)^T\tilde{\Sigma}^{-1}(Y + B\mu - Bx^*) \leq \varepsilon^2, Y \geq \tilde{s}) \\ =&\int_{(y+B\mu-Bx^*)^T\tilde{\Sigma}^{-1}(y+B\mu-Bx^*)\leq\varepsilon^2, y\geq\tilde{s}} \phi(y; 0, \tilde{\Sigma})\mathrm{d}y \\ =&\int_{\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}\leq\varepsilon^2, \tilde{x}\geq\tilde{s}-y^*} \phi(\tilde{x} + y^*; 0, \tilde{\Sigma})\mathrm{d}\tilde{x} \\ =&\left(\prod_{i\in I} e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I\right)^{-1} \int_{\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}\leq\varepsilon^2, \tilde{x}\geq\tilde{s}-y^*} \phi(\tilde{x} + y^*; 0, \tilde{\Sigma})\mathrm{d}x \quad \text{(Similarly, } \tilde{x} \text{ is viewed as a function of } x.) \\ =&(2\pi)^{-\frac{d}{2}}|\tilde{\Sigma}|^{-\frac{1}{2}}\left(\prod_{i\in I} e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I\right)^{-1} \exp\{-(y^*)^T\tilde{\Sigma}^{-1}y^*/2\} \int_{\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}\leq\varepsilon^2, \tilde{x}\geq\tilde{s}-y^*} \exp\{-\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}/2 - x_I^T\mathbf{1}_I\}dx \\ \geq&(2\pi)^{-\frac{d}{2}}|\tilde{\Sigma}|^{-\frac{1}{2}}\left(\prod_{i\in I} e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I\right)^{-1} \exp\{-(y^*)^T\tilde{\Sigma}^{-1}y^*/2\}(1 - \varepsilon^2/2) \int_{\tilde{x}^T\tilde{\Sigma}^{-1}\tilde{x}\leq\varepsilon^2, \tilde{x}\geq\tilde{s}-y^*} \exp\{-x_I^T\mathbf{1}_I\}\mathrm{d}x \\ \geq&(2\pi)^{-\frac{d}{2}}|\tilde{\Sigma}|^{-\frac{1}{2}}\left(\prod_{i\in I} e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I\right)^{-1} \exp\{-(y^*)^T\tilde{\Sigma}^{-1}y^*/2\}(1 - \varepsilon^2/2) \int_{0\leq\tilde{x}\leq\varepsilon'\mathbf{1}} \exp\{-x_I^T\mathbf{1}_I\}\mathrm{d}x \\ =&(2\pi)^{-\frac{d}{2}}|\tilde{\Sigma}|^{-\frac{1}{2}}(1 - \varepsilon^2/2)\varepsilon'^{|J|}\left(\prod_{i\in I}\frac{1 - \exp\{-e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I\varepsilon'\}}{e_i^T(\tilde{\Sigma}_{II})^{-1}\tilde{s}_I}\right)\exp\{-(y^*)^T\tilde{\Sigma}^{-1}y^*/2\} \\ \overset{poly}{\sim}&\exp\{-(y^*)^T\tilde{\Sigma}^{-1}y^*/2\} \\ =&\exp\{-(x^* - \mu)^T\Sigma^{-1}(x^* - \mu)/2\}. \end{aligned}$$

Combining the upper and lower bound for $P(X \in P_1)$, we finally get that

$$P(X \in P_1) \overset{poly}{\sim} \exp\{-(x^* - \mu)^T\Sigma^{-1}(x^* - \mu)/2\}.$$

□

Now we use the asymptotic result in Lemma 7.2 to prove Theorem 4.2.

Proof of Theorem 4.2. Suppose that $g(x) = g_i(x)$ for $h_{ij}(x) \geq 0, j = 1, \ldots, m_i, i = 1, \ldots, r'$ where g_i's and h_{ij}'s are all affine functions. Then we can split $\{x \in \mathbb{R}^d : g(x) \geq \gamma\}$ into $\tilde{\mathcal{R}}_1, \ldots, \tilde{\mathcal{R}}_{r'}$ where $\tilde{\mathcal{R}}_i = \{x \in \mathbb{R}^d : g_i(x) \geq \gamma, h_{ij}(x) \geq 0, j = 1, \ldots, m_i\}$. Without loss of generality, we assume that $P(X \in \tilde{\mathcal{R}}_i) > 0, \forall i = 1, \ldots, r'$ for any $\gamma \in \mathbb{R}$. We denote $\tilde{a}_i = \arg\min\{(x-\mu)^T\Sigma^{-1}(x-\mu) : x \in \tilde{\mathcal{R}}_i\}$. Applying Lemma 7.2, we get that for any $i = 1, \ldots, r'$,

$$P(X \in \tilde{\mathcal{R}}_i) \overset{poly}{\sim} \exp\{-(\tilde{a}_i - \mu)^T\Sigma^{-1}(\tilde{a}_i - \mu)/2\}.$$

Then we get that

$$\tilde{E}[Z] = P(g(X) \geq \gamma) = \sum_{i=1}^{r'} P(X \in \tilde{\mathcal{R}}_i) \overset{poly}{\sim} \exp\{-\min_{i=1,\ldots,r'}(\tilde{a}_i-\mu)^T\Sigma^{-1}(\tilde{a}_i-\mu)/2\} = \exp\{-(a_1-\mu)^T\Sigma^{-1}(a_1-\mu)/2\}. \quad (13)$$

On the other hand, we have that for any $i = 1, \ldots, r$ and $x \in \mathcal{R}_i \subset \{x \in \mathbb{R}^d : (a_i - \mu)^T\Sigma^{-1}(x - a_i) \geq 0\}$,

$$\begin{aligned} L(x) &\leq \frac{re^{-(x-\mu)^T\Sigma^{-1}(x-\mu)/2}}{e^{-(x-a_i)^T\Sigma^{-1}(x-a_i)/2}} \\ &= r\exp\{-(a_i-\mu)^T\Sigma^{-1}(a_i-\mu)/2 - (a_i-\mu)^T\Sigma^{-1}(x-a_i)\} \\ &\leq r\exp\{-(a_i-\mu)^T\Sigma^{-1}(a_i-\mu)/2\} \\ &\leq r\exp\{-(a_1-\mu)^T\Sigma^{-1}(a_1-\mu)/2\}. \end{aligned}$$

Then we get that

$$\tilde{E}[Z^2] = \tilde{E}[I(g(\tilde{X}) \geq \gamma)L^2(\tilde{X})] = E[I(g(X) \geq \gamma)L(X)] \leq r\exp\{-(a_1-\mu)^T\Sigma^{-1}(a_1-\mu)/2\}P(g(X) \geq \gamma). \quad (14)$$

Combining (13) and (14), we finally get that $\tilde{E}[Z^2]/\tilde{E}[Z]^2$ grows at most polynomially growing in γ, and hence Z is asymptotically optimal. □

Proof of Corollary 4.3. See (13) in the proof of Theorem 4.2. □

Proof of Corollary 4.4. Now we suppose that $X \sim f(x) = \sum_{j=1}^m \pi_j\phi(x; \mu_j, \Sigma_j)$. We know that

$$\tilde{E}[Z] = P(g(X) \geq \gamma) = \sum_{j=1}^m \pi_j P(g(X) \geq \gamma | X \sim N(\mu_j, \Sigma_j))$$

and thus from (13),

$$P(g(X) \geq \gamma) \overset{poly}{\sim} \exp\{-\min_{j=1,\ldots,m}(a_{j1} - \mu_j)^T\Sigma_j^{-1}(a_{j1} - \mu_j)/2\}.$$

Moreover, from (14),

$$\begin{aligned}
\tilde{E}[Z^2] &= E[I(g(X) \geq \gamma)L(X)] \\
&= \sum_{j=1}^{m} \pi_j E[I(g(X) \geq \gamma)L(X)|X \sim N(\mu_j, \Sigma_j)] \\
&\leq \sum_{j=1}^{m} \pi_j r_j \exp\{-(a_{j1} - \mu_j)^T \Sigma_j^{-1}(a_{j1} - \mu_j)/2\} P(g(X) \geq \gamma|X \sim N(\mu_j, \Sigma_j)) \\
&\leq \max_{j=1,\ldots,m}\{r_j\} \exp\{-\min_{j=1,\ldots,m}(a_{j1} - \mu_j)^T \Sigma_j^{-1}(a_{j1} - \mu_j)/2\} \sum_{j=1}^{m} \pi_j P(g(X) \geq \gamma|X \sim N(\mu_j, \Sigma_j)) \\
&= \max_{j=1,\ldots,m}\{r_j\} \exp\{-\min_{j=1,\ldots,m}(a_{j1} - \mu_j)^T \Sigma_j^{-1}(a_{j1} - \mu_j)/2\} P(g(X) \geq \gamma).
\end{aligned}$$

Combining the results, we get that $\tilde{E}[Z^2]/\tilde{E}[Z]^2$ at most grows polynomially in γ and the IS estimator Z is asymptotically optimal. □

8 CONCLUSION

In this paper, we study rare-event simulation problems motivated from robustness certification and safety-critical applications of intelligent physical systems, which involve rare-event boundaries associated with the predictions from machine learning models. We consider especially two common predictors, random forest and neural network, and the probability of prediction exceeding a threshold that relates to or forms a building block for the motivating applications. These problems amount to rare-event simulation with piecewise linear set boundaries that are implicitly defined. Our approach merges IS schemes based on the dominating point machinery with sequential integer programming to search for these points in a manner that caters to the geometry of these rare-event sets. We develop asymptotic optimality guarantees, and demonstrate through numerical examples the efficiency of our proposed schemes. Our study can be viewed as a first step to bridge rigorous efficiency-guaranteed rare-event simulation with the emerging applications of AI and intelligent systems. Much warranted further studies include the generalization of our approach to more sophisticated rare-event sets with intricate interaction behaviors, the handling of high-dimensional problems, and the investigation on the impacts of model errors in affecting rare-event probability estimation.

A APPENDIX: TRAINING DETAILS FOR RANDOM FORESTS

In our experiments in Section 6, the random forests are trained using built-in functions in MATLAB. For the regression tasks in Section 6.1, we use the "fitcensemle" function with default setting for training random forests. The function uses bagging (also known as bootstrap aggregating) to train decision trees and ensembles them by averaging their outputs. In particular, each time we train a decision tree, a subset of the input variables is randomly selected as the inputs for prediction and a training set is resampled from the empirical distribution of the original dataset. We use mean squared error as the criterion for branching in training a single decision tree. For the classification tasks in Section 6.2, we use the "fitrensemle" function, which uses boosting to ensemble decision trees trained using the Gini impurity score as a criterion for branching. The function starts with training a relatively small decision tree and then sequentially reduces the prediction error by ensembling new trees that are trained to emphasize the misclassified samples. For more details, please refer to [30, 65].

ACKNOWLEDGEMENTS

We gratefully acknowledge support from the National Science Foundation under grants CAREER CMMI-1653339/1834710, IIS-1849280, IIS-1849304, and the Manufacturing Futures Initiative at Carnegie Mellon University. A preliminary conference version of this work has appeared in [70].

REFERENCES

[1] Robert J Adler, Jose H Blanchet, Jingchen Liu, et al. Efficient monte carlo for high excursions of gaussian random fields. *The Annals of Applied Probability*, 22(3):1167–1214, 2012.

[2] T. P. I. Ahamed, V. S. Borkar, and S. Juneja. Adaptive importance sampling technique for Markov chains using stochastic approximation. *Operations Research*, 54:489–504, 2006.

[3] Dohyun Ahn and Kyoung-Kuk Kim. Efficient simulation for expectations over the union of half-spaces. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 28(3):1–20, 2018.

[4] Mansur Arief, Zhiyuan Huang, Guru Koushik Senthil Kumar, Yuanlu Bai, Shengyi He, Wenhao Ding, Henry Lam, and Ding Zhao. Deep probabilistic accelerated evaluation: A robust certifiable rare-event simulation methodology for black-box safety-critical systems. In *International Conference on Artificial Intelligence and Statistics*, pages 595–603. PMLR, 2021.

[5] S. Asmussen. Conjugate processes and the simulation of ruin problems. *Stochastic Processes and their Applications*, 20:213–229, 1985.

[6] S. Asmussen and K. Binswanger. Simulation of ruin probabilities for subexponential claims. *Astin Bulletin*, 27:297–318, 1997.

[7] S. Asmussen and D. Kroese. Improved algorithms for rare event simulation with heavy tails. *Advances in Applied Probability*, 38:545–558, 2006.

[8] Søren Asmussen and Hansjörg Albrecher. *Ruin probabilities*, volume 14. World scientific, 2010.

[9] Søren Asmussen and Peter W Glynn. *Stochastic Simulation: Algorithms and Analysis*, volume 57. Springer Science & Business Media, New York, 2007.

[10] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[11] M. Bayati, J. Kim, and A. Saberi. A sequential algorithm for generating random graphs. *Approximation, Randomization and combinatorial Optimization. Algorithms and Techniques. Lecture Notes in Computer Science*, 4627:326–340, 2007.

[12] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[13] J. Blanchet. Efficient importance sampling for binary contingency tables. *Annals of Applied Probability*, 19:949–982, 2009.

[14] J. Blanchet, P. Glynn, and H. Lam. Rare event simulation for a slotted time $M/G/s$ model. *Queueing Systems*, 63:33–57, 2009.

[15] J. Blanchet and M. Mandjes. Rare event simulation for queues. In *Rare Event Simulation Using Monte Carlo Methods*, pages 87–124. 2009. Chapter 5.

[16] Jose Blanchet and Peter Glynn. Efficient rare-event simulation for the maximum of heavy-tailed random walks. *The Annals of Applied Probability*, pages 1351–1378, 2008.

[17] Jose Blanchet and Henry Lam. State-dependent importance sampling for rare-event simulation: An overview and recent advances. *Surveys in Operations Research and Management Science*, 17(1):38–59, 2012.

[18] Jose Blanchet and Henry Lam. Rare-event simulation for many-server queues. *Mathematics of Operations Research*, 39(4):1142–1178, 2014.

[19] Jose Blanchet, Henry Lam, and Bert Zwart. Efficient rare-event simulation for perpetuities. *Stochastic Processes and their Applications*, 122(10):3361–3392, 2012.

[20] Jose H Blanchet and Jingchen Liu. State-dependent importance sampling for regularly varying random walks. *Advances in Applied Probability*, 40(4):1104–1128, 2008.

[21] National Transpotation Safety Board. Preliminary report, highway, hwy18mh010, 2018.

[22] National Transpotation Safety Board. Collision between car operating with partial driving automation and truck-tractor semitrailer delray beach, florida, march 1, 2019, 2019.

[23] RK Bock, A Chilingarian, M Gaug, F Hakl, Th Hengstebeck, M Jiřina, J Klaschka, E Kotrč, P Savickỳ, S Towers, et al. Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 516(2-3):511–528, 2004.

[24] P. T. De Boer, V.F. Nicola, and R.Y. Rubinstein. Adaptive importance sampling simulation of queueing networks. In *Proceedings of 2000 Winter Simulation Conference*, pages 646–655. IEEE Press, 2000.

[25] Pierre Bonami, Mustafa Kilinç, and Jeff Linderoth. Algorithms and software for convex mixed integer nonlinear programs. In *Mixed integer nonlinear programming*, pages 1–39. Springer, 2012.

[26] V. S. Borkar, S. Juneja, and A. A. Kherani. Performance analysis conditioned on rare events: An adaptive simulation scheme. *Communications in Information*, 3:259–278, 2004.

[27] Zdravko I Botev and Pierre LâĂŹEcuyer. Sampling conditionally on a rare event via generalized splitting. *INFORMS Journal on Computing*, 2020.

[28] Zdravko I Botev, Pierre LâĂŹEcuyer, and Bruno Tuffin. Markov chain importance sampling with applications to rare event probability estimation. *Statistics and Computing*, 23(2):271–285, 2013.

[29] Fernanda Bravo, Cynthia Rudin, Yaron Shaposhnik, and Yuting Yuan. Simple rules for predicting congestion risk in queueing systems: Application to icus. *Available at SSRN 3384148*, 2019.

[30] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[31] James Bucklew. *Introduction to Rare Event Simulation*. Springer Science & Business Media, New York, 2013.

[32] Joshua CC Chan and Dirk P Kroese. Improved cross-entropy method for estimation. *Statistics and Computing*, 22(5):1031–1040, 2012.

[33] Bohan Chen, Jose Blanchet, Chang-Han Rhee, and Bert Zwart. Efficient rare-event simulation for multiple jump events in regularly varying random walks and compound poisson processes. *Mathematics of Operations Research*, 44(3):919–942, 2019.

[34] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730, 2015.

[35] Zhilu Chen and Xinming Huang. End-to-end learning for lane keeping of self-driving cars. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1856–1860. IEEE, 2017.

[36] Leon O Chua and Lin Yang. Cellular neural networks: Theory. *IEEE Transactions on circuits and systems*, 35(10):1257–1272, 1988.

[37] Jeffrey F Collamore. Importance sampling techniques for the multidimensional ruin problem for general markov additive sequences of random vectors. *The Annals of Applied Probability*, 12(1):382–421, 2002.

[38] P. T. de Boer, D. Kroese, S. Mannor, and R. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134:19–67, 2005.

[39] Thomas Dean and Paul Dupuis. Splitting for rare event simulation: A large deviation approach to design and analysis. *Stochastic Processes and their Applications*, 119(2):562 – 587, 2009.

[40] Anand Deo and Karthyek Murthy. Efficient black-box importance sampling for var and cvar estimation. *arXiv preprint arXiv:2106.10236*, 2021.

[41] P. Y. Desai and P. W. Glynn. A Markov chain perspective on adaptive Monte Carlo algorithms. *Proceedings of 2001 Winter Simulation Conference*, 9:391–412, 2001.

[42] AB Dieker and Michel Mandjes. On asymptotically efficient simulation of large deviation probabilities. *Advances in applied probability*, 37(2):539–552, 2005.

[43] Tommaso Dreossi, Alexandre Donzé, and Sanjit A Seshia. Compositional falsification of cyber-physical systems with machine learning components. *Journal of Automated Reasoning*, 63(4):1031–1053, 2019.

[44] Paul Dupuis, Kevin Leder, and Hui Wang. Importance sampling for sums of random variables with regularly varying tails. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 17(3):14–es, 2007.

[45] Paul Dupuis, Kevin Leder, and Hui Wang. Importance sampling for weighted-serve-the-longest-queue. *Mathematics of Operations Research*, 34(3):642–660, 2009.

[46] Paul Dupuis, Konstantinos Spiliopoulos, and Hui Wang. Importance sampling for multiscale diffusions. *Multiscale Modeling & Simulation*, 10(1):1–27, 2012.

[47] Jakub Dvořák and Petr Savickỳ. Softening splits in decision trees using simulated annealing. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 721–729. Springer, 2007.

[48] Laura Fraade-Blanar, Marjory S Blumenthal, James M Anderson, and Nidhi Kalra. *Measuring automated vehicle safety: forging a framework*. RAND Corporation, 2018.

[49] Hongbo Gao, Bo Cheng, Jianqiang Wang, Keqiang Li, Jianhui Zhao, and Deyi Li. Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment. *IEEE Transactions on Industrial Informatics*, 14(9):4224–4231, 2018.

[50] Roy Glasius, Andrzej Komoda, and Stan CAM Gielen. Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, 8(1):125–133, 1995.

[51] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. Multilevel splitting for estimating rare event probabilities. *Operations Research*, 47:585–600, 1999.

[52] Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.

[53] Paul Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53. Springer Science & Business Media, New York, 2013.

[54] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. A large deviations perspective on the efficiency of multilevel splitting. *IEEE Transactions on Automatic Control*, 43(12):1666–1679, 1998.

[55] Paul Glasserman, Wanmo Kang, and Perwez Shahabuddin. Fast simulation of multifactor portfolio credit risk. *Operations Research*, 56(5):1200–1217, 2008.

[56] Paul Glasserman and Jingyi Li. Importance sampling for portfolio credit risk. *Management Science*, 51(11):1643–1656, 2005.

[57] Paul Glasserman and Yashan Wang. Counterexamples in importance sampling for large deviations probabilities. *The Annals of Applied Probability*, 7(3):731–746, 1997.

[58] Peter W Glynn and Donald L Iglehart. Importance sampling for stochastic simulations. *Management Science*, 35(11):1367–1392, 1989.

[59] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep Learning*, volume 1. MIT press Cambridge, Massachusetts, 2016.

[60] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[61] Adam W Grace, Dirk P Kroese, and Werner Sandmann. Automated state-dependent importance sampling for markov jump processes via sampling from the zero-variance distribution. *Journal of Applied Probability*, 51(3):741–755, 2014.

[62] P. Grassberger. Go with the winners: A general Monte Carlo strategy. *Computer Physics Communications*, 147:64–70, 2002.

[63] Matthew S Hagen, Jeffrey K Jopling, Timothy G Buchman, and Eva K Lee. Priority queuing models for hospital intensive care units and impacts to severe case patients. In *AMIA Annual Symposium Proceedings*, volume 2013, page 841. American Medical Informatics Association, 2013.

[64] Enkelejd Hashorva and Juerg Huesler. On multivariate gaussian tails. *Annals of the Institute of Statistical Mathematics*, 55:507–522, 02 2003.

[65] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Random forests. In *The elements of statistical learning*, pages 587–604. Springer, 2009.

[66] Hera Y. He and Art B. Owen. Optimal mixture weights in multiple importance sampling. nov 2014.

[67] P. Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 5:43–85, 1995.

[68] Harsha Honnappa, Raghu Pasupathy, and Prateek Jaiswal. Dominating points of gaussian extremes, 2018.

[69] Z. Huang, H. Lam, D. J. LeBlanc, and D. Zhao. Accelerated evaluation of automated vehicles using piecewise mixture models. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2017.

[70] Zhiyuan Huang, Henry Lam, and Ding Zhao. Designing importance samplers to simulate machine learning predictors via optimization. In *2018 Winter Simulation Conference (WSC)*, pages 1730–1741. IEEE, 2018.

[71] Zhiyuan Huang, Henry Lam, and Ding Zhao. Rare-event simulation without structural information: a learning-based approach. In *2018 Winter Simulation Conference (WSC)*, pages 1826–1837. IEEE, 2018.

[72] Henrik Hult and Jens Svensson. On importance sampling with mixtures for random walks with heavy tails. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 22(2):1–21, 2012.

[73] S. Juneja and P. Shahabuddin. Chapter 11 rare-event simulation techniques: An introduction and recent advances. In Shane G. Henderson and Barry L. Nelson, editors, *Simulation*, volume 13 of *Handbooks in Operations Research and Management Science*, pages 291 – 350. Elsevier, 2006.

[74] Sandeep Juneja and Perwez Shahabuddin. Rare-event simulation techniques: An introduction and recent advances. *Handbooks in Operations Research and Management Science*, 13:291–350, 2006.

[75] Nidhi Kalra and Susan M Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[76] G. Kesidis, J. Walrand, and C.-S. Chang. Effective bandwidths for multiclass Markov fluids and other ATM sources. *IEEE/ACM Transactions on Networks.*, 1:424–428, 1993.

[77] Mykel J Kochenderfer, Jessica E Holland, and James P Chryssanthacopoulos. Next-generation airborne collision avoidance system. Technical report, Massachusetts Institute of Technology-Lincoln Laboratory Lexington United States, 2012.

[78] C. Kollman, K. Baggerly, D. Cox, and R. Picard. Adaptive importance sampling on discrete Markov chains. *Annals of Applied Probability*, 9:391–412, 1999.

[79] Philip Koopman and Michael Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2017.

[80] D. P. Kroese and V. F. Nicola. Efficient estimation of overflow probabilities in queues with breakdowns. *Performance Evaluation*, 36-37:471–484, 1999.

[81] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[82] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[83] Kristofer D Kusano and Hampton C Gabler. Safety benefits of forward collision warning, brake assist, and autonomous braking systems in rear-end collisions. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1546–1555, 2012.

[84] P. L'Ecuyer, F. Le Gland, P. Lezaud, and B. Tuffin. Splitting techniques. In *Rare Event Simulation Using Monte Carlo Methods*, pages 39–62. 2009. Chapter 3.

[85] Pierre L'Ecuyer, Jose H. Blanchet, Bruno Tuffin, and Peter W. Glynn. Asymptotic robustness of estimators in rare-event simulation. *ACM Trans. Model. Comput. Simul.*, 20(1), February 2010.

[86] Velibor V Mišic. Optimization of tree ensembles. *arXiv preprint arXiv:1705.10883*, 2017.

[87] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann L Cun. Off-road obstacle avoidance through end-to-end learning. In *Advances in neural information processing systems*, pages 739–746, 2006.

[88] Karthyek RA Murthy, Sandeep Juneja, and Jose Blanchet. State-independent importance sampling for random walks with regularly varying increments. *Stochastic Systems*, 4(2):321–374, 2015.

[89] Victor F Nicola, Marvin K Nakayama, Philip Heidelberger, and Ambuj Goyal. Fast simulation of highly dependable systems with general failure and repair processes. *IEEE Transactions on Computers*, 42(12):1440–1452, 1993.

[90] Victor F Nicola, Perwez Shahabuddin, and Marvin K Nakayama. Techniques for fast simulation of models of highly dependable systems. *IEEE Transactions on Reliability*, 50(3):246–264, 2001.

[91] Matthew O'Kelly, Aman Sinha, Hongseok Namkoong, Russ Tedrake, and John C Duchi. Scalable end-to-end autonomous vehicle testing via rare-event simulation. In *Advances in Neural Information Processing Systems*, pages 9827–9838, 2018.

[92] Art B Owen, Yury Maximov, Michael Chertkov, et al. Importance sampling the union of rare events with an application to power systems analysis. *Electronic Journal of Statistics*, 13(1):231–254, 2019.

[93] P. Glasserman, P. Heidelberger and P. Shahabuddin and T. Zajic. A large deviations perspective on the effiency of multilevel splitting. *IEEE Transactions on Automated Control*, pages 1666–1679, 1998.

[94] S. Parekh and J. Walrand. Quick simulation of rare events in networks. *IEEE Transactions on Automatic Control*, 34:54–66, 1989.

[95] Fait Poms, Vishnu Sarukkai, Ravi Teja Mullapudi, Nimit S Sohoni, William R Mark, Deva Ramanan, and Kayvon Fatahalian. Low-shot validation: Active importance sampling for estimating classifier performance on rare categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10705–10714, 2021.

[96] R. Y. Rubinstein. Rare-event simulation via cross-entropy and importance sampling. *Second Workshop on Rare Event Simulation, RESIMâĂŹ99*, pages 1–17, 1999.

[97] Martin Rajchl, Matthew CH Lee, Ozan Oktay, Konstantinos Kamnitsas, Jonathan Passerat-Palmbach, Wenjia Bai, Mellisa Damodaram, Mary A Rutherford, Joseph V Hajnal, Bernhard Kainz, et al. Deepcut: Object segmentation from bounding box annotations using convolutional neural networks. *IEEE transactions on medical imaging*, 36(2):674–683, 2016.

[98] A. Ridder. Importance sampling algorithms for first passage time probabilities in the infinite server queue. *European Journal of Operational Research*, 199:176–186, 2009.

[99] G. Rubino and B. Tuffin. Markovian models for dependability analysis. In *Rare Event Simulation Using Monte Carlo Methods*, pages 125–144. 2009. Chapter 6.

[100] R. Rubinstein and D. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer-Verlag, 2004.

[101] R. Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operations Research*, 99:89–112, 1997.

[102] Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo Method*, volume 10. John Wiley & Sons, New Jersey, 2016.

[103] J. S. Sadowsky. Large deviations theory and efficient simulation of excessive backlogs in a $GI/GI/m$ queue. *IEEE Transactions on Automatic Control*, 36:1383–1394, 1991.

[104] John S Sadowsky and James A Bucklew. On large deviations theory and asymptotically efficient monte carlo estimation. *IEEE transactions on Information Theory*, 36(3):579–588, 1990.

[105] W. Sandmann. Rare event simulation methodologies in systems biology. In *Rare Event Simulation Using Monte Carlo Methods*, pages 243–266. 2009. Chapter 11.

[106] Petr Savickỳ and Emil Kotrc. Experimental study of leaf confidences for random forest. In *Proceedings of the 16th Symposium on Computational Statistics*, pages 1767–1774. Prague, Czech Republic, 2004.

[107] Christoph Sawade, Niels Landwehr, and Tobias Scheffer. Active estimation of f-measures. *Advances in Neural Information Processing Systems*, 23:2083–2091, 2010.

[108] David Siegmund. Importance sampling in the monte carlo study of sequential tests. *The Annals of Statistics*, pages 673–684, 1976.

[109] Nathan A Spielberg, Matthew Brown, Nitin R Kapania, John C Kegelman, and J Christian Gerdes. Neural network vehicle models for high-performance automated driving. *Science Robotics*, 4(28), 2019.

[110] R. Szechtman and P. Glynn. Rare event simulation for infinite server queues. In *Proceedings of the 2002 Winter Simulation Conference*, pages 416–423, 2002.

[111] Vincent Tjeng and Russ Tedrake. Verifying neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

[112] B. Tuffin. On numerical problems in simulation of highly reliable Markovian systems. In *Proceedings of the 1st International Conference on Quantitative Evaluation of SysTems (QEST)*, pages 156–164. IEEE Computer Society Press, 2004.

[113] Jonathan Uesato, Ananya Kumar, Csaba Szepesvari, Tom Erez, Avraham Ruderman, Keith Anderson, Nicolas Heess, Pushmeet Kohli, et al. Rigorous agent evaluation: An adversarial approach to uncover catastrophic failures. *arXiv preprint arXiv:1812.01647*, 2018.

[114] Jessica Van Brummelen, Marie OâĂŹBrien, Dominique Gruyer, and Homayoun Najjaran. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation research part C: emerging technologies*, 89:384–406, 2018.

[115] C Van Rijsbergen. Information retrieval: theory and practice. In *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*, pages 1–14, 1979.

[116] Eric Vanden-Eijnden and Jonathan Weare. Rare event simulation of small noise diffusions. *Communications on Pure and Applied Mathematics*, 65(12):1770–1803, 2012.

[117] Stephen A. Vavasis. Complexity Theory: Quadratic Programming. In *Encyclopedia of Optimization*, pages 304–307. Springer US, jun 2006.

[118] Benjie Wang, Stefan Webb, and Tom Rainforth. Statistically robust neural network classification. *arXiv preprint arXiv:1912.04884*, 2019.

[119] Stefan Webb, Tom Rainforth, Yee Whye Teh, and M Pawan Kumar. A statistical approach to assessing neural network robustness. *arXiv preprint arXiv:1811.07209*, 2018.

[120] Tsui-Wei Weng, Pin-Yu Chen, Lam M Nguyen, Mark S Squillante, Ivan Oseledets, and Luca Daniel. Proven: Certifying robustness of neural networks with a probabilistic approach. *arXiv preprint arXiv:1812.08329*, 2018.

[121] Jianxin Wu, James M Rehg, and Matthew D Mullin. Learning a rare event detection cascade by direct feature selection. In *Advances in Neural Information Processing Systems*, pages 1523–1530, 2004.

[122] Simon X Yang and Chaomin Luo. A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):718–724, 2004.

[123] Ding Zhao, Xianan Huang, Huei Peng, Henry Lam, and David J. LeBlanc. Accelerated evaluation of automated vehicles in car-following maneuvers. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):733–744, 2018.

[124] Ding Zhao, Henry Lam, Huei Peng, Shan Bao, David J LeBlanc, Kazutoshi Nobukawa, and Christopher S Pan. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE transactions on intelligent transportation systems*,

18(3):595–607, 2016.