IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 16, 2021, accepted August 11, 2021, date of publication August 16, 2021, date of current version August 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3105555

Attack Transferability Against Information-Theoretic Feature Selection

SRISHTI GUPTA, ROMAN GOLOTA, AND GREGORY DITZLER, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721, USA

Corresponding author: Gregory Ditzer (ditzler@arizona.edu)

This work was supported in part by the Department of Energy under Grant DE-NA0003946, in part by the Army Research Office under Grant W56KGU-20-C-0002, in part by the Air Force Office of Scientific Research under Grant FA8750-19-C-0209, and in part by the National Science Foundation CAREER under Grant 1943552.

ABSTRACT Machine learning (ML) is vital to many application-driven fields, such as image and signal classification, cyber-security, and health sciences. Unfortunately, many of these fields can easily have their training data tampered with by an adversary to thwart an ML algorithm's objective. Further, the adversary can impact any stage in an ML pipeline (e.g., preprocessing, learning, and classification). Recent work has shown that many models can be attacked by poisoning the training data, and the impact of the poisoned data can be quite significant. Prior works on adversarial feature selection have shown that the attacks can damage feature selection (FS). Filter FS algorithms, a type of FS, are widely used for their ability to model nonlinear relationships, classifier independence and lower computational requirements. One important question from the security perspective of these widely used approaches is, whether filter FS algorithms are robust against other FS attacks. In this work, we focus on the task of information-theoretic filter FS such MIM, MIFS, and mRMR, and the impact that gradient-based attack can have on these selections. The experiments on five benchmark datasets demonstrate that the stability of different information-theoretic algorithms can be significantly degraded by injecting poisonous data into the training dataset.

INDEX TERMS Adversarial machine learning, feature selection, information theory.

I. INTRODUCTION

Machine learning has transformed application-driven fields and despite its prevalence, the security and privacy of these techniques are easily compromised when there is an adversary in the environment [1]–[6]. Recent works have shown that shallow/deep neural networks [7], [8], support vector machines [9], [10], feature selection [11]–[13] and generalized linear models are all subject to attacks that can be launched at training or testing time. For example, recent findings have shown how easy it is to fool a classifier by having the adversary change the context of spam emails to ham that read nearly the same. This issue of security is serious if data only need to be slightly modified to drastically change the classifier's output, which ultimately begs the question of the stability of FS algorithm used in the model. This is an important question to address because FS is used in nearly all data science pipelines and remains a critical aspect of exploratory data analysis. Therefore, the community needs to understand how FS is influenced by an adversary and the degree to which different FS techniques can be influenced.

FS algorithms fall into one of three categories: *wrapper-*, *embedded-*, and *filter-based* approaches [14]. Wrapper-based FS algorithms optimize the feature set for a specific classifier, and, therefore, these methods tend to require large amounts of computational resources, which makes them infeasible for many datasets. Embedded-based FS algorithms optimize the parameters of a classifier and feature selector simultaneously. Both the embedded and wrapper methods are dependent on the optimization of a classifier or dependent upon the classifier that is selected. Filter-based FS algorithms score features by a function to determine importance independent of a classifier's error. The advantage of a filter-based approach is that they have typically much lower computational complexity than wrapper or embedded methods. In this work, we specifically focus on filter methods because they are classifier independent, which allows them a bit more flexible in their application. Further, another reason for focusing on filter methods is that Brown *et al.* showed that users can optimize the feature set and classifier independently with

The associate editor coordinating the review of this manuscript and approving it for publication was Joanna Kołodziej.

the filter assumption [15]. While adversarial learning has a traditional focus on attacks against the classifier, far fewer works exist to address how the adversary can influence FS.

Adversarial learning focuses on (a) generating classifiers that are more robust to attacks at the testing time (*evasion attacks*), or (b) generating adversarial samples that negatively impact the training of a classifier (*poisoning attacks*) [16]. The adversary can manipulate data to achieve their goal of bypassing a machine learning classifier by forcing it to make an error. The process of deceiving the classifier becomes complicated when data are tampered prior to the preprocessing stage. Feature selection, which is an essential preprocessing step in many machine learning pipelines, is done to reduce the impact of the curse of dimensionality on a classifier's generalization performance [14].

Unfortunately, contributions to adversarial FS are still somewhat limited and remain an important topic because of feature selection's role in nearly all data science pipelines. Considering FS with adversaries is also essential for applied research in cyber-security where FS is used, but the adversary is not taken into account [4], [17]–[21]. Therefore, it is essential to examine what we know about an adversary's impact on FS in the same way we know about its impact on classification.

There are several reasons that an adversary would want to impact the outcome of FS. For example, consider a dataset that has ten features $(X_1, \ldots, X_{10})$, and we are tasked to select the three most informative features from the original dataset without an adversary (e.g., let X_1, X_2 and X_3 are the best features in the dataset). Consider that an adversary can inject samples into the training dataset to make X_3 not appear informative. It is not because X_3 is not informative but can be easily manipulated to bypass the system by the adversary. An example of such an application is in cyber-security, where removing a variable from the relevant feature set can allow the adversary to launch attacks on files while going undetected [22].

An attack model for *Least Absolute Shrinkage and Selection Operator* (LASSO) was recently presented by Xiao *et al.* in [13] and Tibshirani [23]. Their attack model uses gradient-based methods to directly attack LASSO that can significantly increase the error and decrease the stability (i.e., how repeatable the LASSO features would select). While their work showed that this attack algorithm is easy to implement and very successful against LASSO, it is still unknown how these attacks will transfer to other FS approaches (e.g., filter FS). The notation of "transfer" refers to an attack designed for a specific algorithm then applied the attack to a different algorithm. If the attack still works against a different algorithm, then the attack is considered transferable. This concept is known as transferability, and it is essential to note there are varying degrees of transferability (e.g., the attack is successful 1% or 95% of the time).

In this contribution, we study the transferability of gradient-based attacks against information-theoretic FS, which is a filter-based approach. We present findings that show the impact and transferability of using an information-theoretic FS algorithm against attacks designed for gradient-based FS algorithms. Experiments on five different UCI datasets support the hypothesis that an adversary can negatively impact feature selection without perfect knowledge of the model being used by the practitioner [16].

This paper is organized as follows: Section II summarizes the system-driven taxonomy, Section III highlights problem setup and related works in data preprocessing and machine learning, Section IV describes parameters and nomenclatures used in the experiments. We report experiments on five different datasets and assess how gradient-based attacks perform on information-theoretic FS algorithms in Section V. A discussion and concluding remarks are in Section VI and VII.

II. TAXONOMY

One of the most important pieces to an adversary is the assumptions that they make. We briefly revisit the system-driven taxonomy in adversarial ML practices [16]. The attacker's model is defined by the attributes: 1) *Knowledge*, 2) *Capability*, 3) *Goal*, and 4) *Strategy* (shown in Figure 1). Adversary can have two main types of knowledge: a) *data* and b) *algorithm*. The level of knowledge can be from zero, limited, or full. Zero-Knowledge adversaries launch *or black-box attacks* since they do not know the data, feature selector or classifier. Limited Knowledge, *or gray-box attacks*, are when the adversary has access to some, but not all of the information related to a user's task. Perfect Knowledge *white-box attacks* are those where the adversary has access to everything the user has access, which tend to lead to the most damaging attacks. Adversary's capability depends on the influence it has on input data and data-manipulation constraints. In practical scenarios, an adversary can only alter only a portion of data. Depending on the influence and access to data, an attack can be *poisoning* (or causative) when the attacker alters the training data, or *evasion* (or exploratory) when the attacker alters the testing data. Adversary's goal can depend on what it aims to violate among a) confidentiality, b) integrity, or c) availability [24]. Finally, the attack strategy can either be *Passive* where the adversary aims to derive information about the application/user (often used in evasion attacks), or *Active* where attacks hamper the application/user's normal operation (often used in poisoning attacks).

In this work, the adversary's goal is further explored. What is the goal of the adversary? Is the adversary's intention to directly harm the feature selection algorithm by choosing features that lead to instability, or they want to remove a specific feature from $\mathcal{S}$, set of selected features? In this work, the adversary's knowledge of data and classifier is limited (i.e., they have information about the data; but, they do not know which classifier or feature selection algorithm is used), the adversary has access to training data and the capability to append poisoning samples to the original training dataset. The adversary's goal is to degrade the performance of feature selection by making FS algorithms unstable and

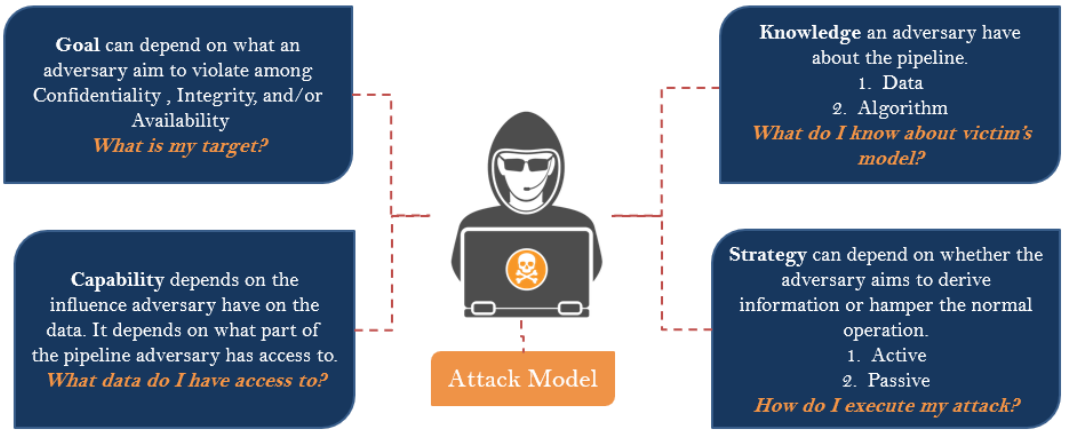


FIGURE 1. Representation of the considerations that goes behind the generation of the attack model.

inconsistent. To achieve this goal, the adversary uses a poisoning attack (LASSO attack) as his/her strategy. We feel that this intention is the most reasonable to study the robustness of information-theoretic FS algorithms because no attacks are directly geared towards information-theoretic FS.

III. PROBLEM SETUP AND NOTATION

In this work, we assume the attacker has the knowledge of data but not of the preprocessing pipeline (e.g., the FS method being used). A dataset D is represented by tuples of feature vectors and corresponding label denoted by $(\mathbf{x}_n, y_n)$ where $\mathbf{x}_n \in \mathbb{R}^P$ is a P-dimensional vector that is a collection of random variables in the set $\mathcal{X} := \{X_1, \ldots, X_P\}$ and the corresponding class label is $y_n \in \{\pm 1\}$. The class label y_n can also be viewed as a random variable Y. In this work, we focus on binary classification tasks and leave multi-class problems as future work. The index n denotes the nth data sample. The objective of FS is to determine a subset of features $\mathcal{S} \subset \mathcal{X}$ that are informative and possibly not redundant. We assume the adversary can poison the model by appending samples to the training dataset D_{tr}. The goal of the adversary is to violate the security of the FS algorithm by launching a causative attack (i.e., an attack meant to inflict damage at training [16]). The FS algorithm's security becomes compromised when the adversary successfully introduce malicious samples into the dataset D^*_{tr} that causes the algorithm to choose different features than if it were run solely on the legitimate (i.e., untampered) training dataset D_{tr}. Choosing different features is on measure of how we can judge a FS algorithm's performance and it is important to understand how the performance of FS is different than that of classification tasks with an adversary.

FS algorithms are not compared to each other using the same figures of merit as a classifier (e.g., F-score, accuracy, AUC, etc.). Rather, FS algorithms are compared using measures of stability and consistency [25], [26]. For example, the Jaccard score is a straightforward similarity measure between two sets that can determine the degree of similarity between two feature sets. In this work, we use measures such as the Jaccard score in two different ways. First, we use FS stability to measure the consistency of an algorithm on a dataset. For example, a FS is run M times on bootstrap datasets. How consistent are these M sets to each other? This is a measure of consistency between the features that were selected from different bootstrap datasets. Note that this measure of performance is independent of an adversary being present or not. Second, we use FS stability measures to quantify the *similarity* between an adversarial and benign feature set. Thus, this measure determines the distance between the features that would have been selected if there were no adversary (i.e., benign) and the features selected when an adversary performed a causative attack. We formally describe these measures in Section VI-B.

IV. RELATED WORK AND BACKGROUND

Selection of the most relevant features towards correct classification and robustness is essential to improve the model's performance and, reduce the computational cost, avoid overfitting. Further, FS can make the model easier to interpret and can be performed in several ways based on the features' interaction with the class label. FS algorithms are categorized as embedded, wrapper or filter methods. The main focus of this work is to study the performance of embedded FS attacks against information-theoretic filter FS techniques. We describe these techniques in this section.

A. EMBEDDED METHODS

Embedded methods select features by jointly optimizing a classification loss and feature selector objective (e.g., l_1 regularization term that induces sparsity in LASSO [23]). LASSO selects features by learning linear function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ that minimizes the trade-off between loss function $l(y, f(\mathbf{x}))$ and regularization term $\Omega(\mathbf{w})$, and can be written as:

$$\arg\min_{\mathbf{w},b} \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(\mathbf{x}_i)) + \lambda\|\mathbf{w}\|_1 \tag{1}$$

where $\lambda > 0$ is a free parameter and $\|\cdot\|_1$ is the l_1-norm. LASSO performs shrinkage (i.e., regularization) to select features. This method applies the shrinkage process to penalize the coefficients of the regression variables shrinking some of them to zero. The non-zero terms remaining in $\mathbf{w}$ are then selected features. λ plays an important role to control the strength of the penalty. Larger values of λ force more coefficients to shrink to zero, causing the reduction in dimensionality. When $\lambda = 0$, model acts as a simple linear classifier without FS.

B. INFORMATION-THEORETIC FEATURE SELECTION

In filter approaches, some examples of evaluation functions are probabilistic distance, inter-class distance, information-theoretic or probabilistic dependence measures [27]. Because these measures are directly calculated on raw data instead of learned model that smooths the noise, they are often considered as intrinsic properties of the data [28]. To implement filter FS methods, many different methods can be used e.g., correlation filters [29], statistical-based, information-theoretic, etc. Information-theoretic methods are quite popular for their speed, and theoretical properties [15], [30], which is also the focus of this paper. Many filter FS methods are based on fundamental information-theoretic quantities (e.g., entropy, mutual information, Kullback-Leibler divergence, etc. [31]). Information-theoretic-based filter FS algorithms rank features using a criterion $J(\cdot)$ [15], referred as "relevancy index" or "scoring function", which quantifies how useful a feature – or subset of features – can be for classification. Greedy forward search algorithms are typically used to find a feature subset of top k features (e.g., 10% of total features) from a dataset. In this section, we review four popular information-theoretic approaches that are used in this work, namely: Mutual Information Maximization, Mutual Information Feature Selection, Minimum Redundancy Maximum Relevancy and Double Input Symmetrical Relevance.

1) MUTUAL INFORMATION MAXIMIZATION (MIM)

MIM is a simple information-theoretic approach that only uses the mutual information (MI) score to rank the feature set. The MI score of each feature X_k with Y is calculated independently. The scores are ranked and the top k features are chosen [32]. MIM's score is given by:

$$J_{MIM}(X_k) = I(X_k; Y) \tag{2}$$

Note J_{MIM} is the notation used to denoted the objective function for MIM. Unfortunately, MIM does not capture the redundancy of features (i.e., two or more features having common information).

2) MUTUAL INFORMATION FEATURE SELECTION (MIFS)

MIFS uses a greedy selection algorithm that takes both relevancy and redundancy into account. The MI score between a feature and output is calculated, similar to MIM; however, there is an additional term in the score function that captures redundancy between the feature under test and the features that have already been selected. The relevancy of a feature X_k and redundancy between two features X_k and X_j is written as

$$J_{MIFS}(X_k) = I(X_k, Y) - \beta \sum_{X_j \in \mathcal{S}} I(X_k; X_j) \tag{3}$$

where $\beta > 0$ free parameter and $\mathcal{S}$ is the set of features that were previously selected.

3) MINIMUM REDUNDANCY MAXIMUM RELEVANCY (mRMR)

mRMR takes relevancy-redundancy criteria one step further by defining an "optimal" subspace of features [33]. The objective function is formally given by:

$$J_{MRMR} = I(X_k, Y) - \frac{1}{|\mathcal{S}|} \sum_{X_j \in \mathcal{S}} I(X_k; X_j) \tag{4}$$

where the scaling term in mRMR is when MIFS has $\beta = 1/|\mathcal{S}|$. mRMR and MIFS are slightly different techniques so it becomes important to understand the effect of attacks generated on embedded methods against the information-theoretic methods.

4) DOUBLE INPUT SYMMETRICAL RELEVANCE (DISR)

DISR combines two well-known properties for FS [28]. First, DISR exploits the concept of variable complementarity. Variable complementarity is when a combination of features returns more information about the output class than the sum of the information returned by each feature individually. Second, computing lower bound on the information of a set of features expressed as the average information of all its subsets. This lower bound approximation significantly reduces the computational overhead and these approximations have been quite successful in FS [34], [35]. DISR ranks the features based on a symmetric relevance score that is the ratio of the mutual information and joint entropy.

C. ATTACK STRATEGY

Attacks are developed based on the amount of information the adversary has about the user's model (e.g., availability of dataset, and adversary's capabilities to generate attacks [36]). This work assumes that the adversary's goal is to violate the availability (i.e., compromising the system's functionality of the user) with partial knowledge of the system. The adversary only knows that the user has a feature selector and classifier in their pipeline, but is unaware of the specific algorithms that are being used. The adversary has access to user's dataset D and has the intent to inject adversarial samples into the training dataset D_{tr}, which will poison the FS algorithm. Given the attacker's limited knowledge of the classifier θ, and their capability Φ to manipulate good samples $\mathcal{A}$ to malicious samples $\mathcal{A}' \in \Phi(\mathcal{A})$, one can calculate attack strength using objective function, $\mathcal{W}(\mathcal{A}', \theta)$. Optimal attack strategy [15] can thus be defined by maximising $\mathcal{W}$ subject to the attacker's

capabilities:

$$\max_{\mathcal{A}'} \ \mathcal{W}(\mathcal{A}'; \theta)$$
$$\text{s.t. } \mathcal{A}' \in \Phi(\mathcal{A}) \tag{5}$$

V. ADVERSARIAL ATTACKS AGAINST FEATURE SELECTION

Recent work by Xiao *et al.* showed that an adversarial attack algorithm can easily be formulated against algorithms such as LASSO, ridge regression, and elastic nets [13], [23], [37]. Xiao *et al.* proposed an attack strategy that reverses the LASSO's cost function, forcing it to select a poor set of features. In their work, LASSO's original objective function, given in Eq. (1), is manipulated to the objective function shown in Eq. (6). Unfortunately, a knowledgeable adversary knows a significant amount of information about user's task that is using LASSO. First, the adversary knows the task the user wants to perform (e.g., binary classification or regression). Second, the adversary might know that the user chose LASSO for FS. If the adversary did not have access to the user's FS algorithm, we need to show the transferability of the attack space. If the adversary uses Xiao *et al.*'s algorithm to generate LASSO attack to poison the user's training data and the user chose an information-theoretic FS method, will the information-theoretic methods fail? At what rate do they fail? The primary contribution of this work is to understand the answers to these two questions, critical to machine learning and data science practitioners.

Let us assume that the attacker decides to use Xiao *et al.*'s attack algorithm (discussed below). Then the adversary fundamentally knows how LASSO works. Therefore, generating adversarial samples to poison the data is performed by casting LASSO's original objective as a task that generates a data sample that increases LASSO's error (i.e., the optimization is not over $\mathbf{w}$ or b). This task is achieved by maximizing the cost function of LASSO. The Xiao et al's attack strategy generates a single attack point $\mathbf{x}_c$ against embedded algorithms. The adversary's objective $\mathcal{W}$ can be written as:

$$\max_{x_c} \mathcal{W} = \frac{1}{m}\sum_{j=1}^{m} l(\hat{y}_j, f(\hat{\mathbf{x}}_j)) + \lambda\Omega(\mathbf{w}) \tag{6}$$

where m is the number of samples in D^*_{tr} with all of the benign plus the adversarial samples and f is a linear function from LASSO with parameters $\mathbf{w}$ and b. Several points are worth noting that contrast LASSO's original formulation. First, the original optimization task was performed over $\mathbf{w}$ or b; however, now we assume those parameters are fixed, and the optimization is over the attack/poison sample $\mathbf{x}_c$. Second, the LASSO optimization was a minimization task; however, the optimization of attack sample $\mathbf{x}_c$ is a maximization task. This conversion to a maximization task is rather easy to understand; however, this formulation has a subtle consequence. The goal is to iteratively maximize $\mathcal{W}$. Then an unbounded $\mathbf{x}_c$ is a trivial solution to the optimization problem.

Algorithm 1 Poisoning Embedded Feature Selection

Input: Training data D_{tr}
Input: $\{\mathbf{x}_c^{t=0}, y_c\}_{c=1}^{q}$, q initial attack points with labels
Input: σ, ϵ small positive constants, $\beta \in (0, 1)$
$t = 0$
repeat
  for $c = 0, \ldots, q$ **do**
    $\{\mathbf{w}, b\} \leftarrow$ learn classifier on $D_{tr} \cup \{\mathbf{x}_c^t\}_{c=1}^{q}$
    Calculate $\nabla\mathcal{W}$ using Eq. (7)
    Set $\mathbf{d} = \Pi_{\mathcal{B}}(\mathbf{x}_c^t + \nabla\mathcal{W}) - \mathbf{x}_c^t$ and $k \leftarrow 0$
    repeat
      Set $\eta \leftarrow \beta^k$ using line search and $k \leftarrow k + 1$
      $\mathbf{x}_c^{t+1} \leftarrow \mathbf{x}_c^t + \eta\mathbf{d}$
    until $\mathcal{W}(\mathbf{x}_c^{t+1}) \leq \mathcal{W}(\mathbf{x}_c^t) - \sigma\eta\|\mathbf{d}\|^2$;
  $t = t + 1$
until $|\mathcal{W}(\{x_c^t\}_{c=1}^{q}) - \mathcal{W}(\{x_c^{t-1}\})_{c=1}^{q}| < \epsilon$;
return $\{\mathbf{x}_c^t\}_{c=1}^{q}$

Therefore, the authors of [13] place a bounding box on $\mathbf{x}_c$ to prevent the solution from approaching infinity.

Solving Eq. (6) can be done with a straightforward gradient ascent algorithm that was presented in Algorithm 1. In order, to solve Eq. (6), we need to calculate the gradient w.r.t. $\mathbf{x}_c$ which is given by:

$$\frac{\partial\mathcal{W}}{\partial\mathbf{x}_c} = \frac{1}{m}\sum_{j=1}^{m}\left(f(\hat{\mathbf{x}}_j) - \hat{y}_j\right)\left(\hat{\mathbf{x}}_j^{\intercal}\frac{\partial\mathbf{w}}{\partial\mathbf{x}_c} + \frac{\partial b}{\partial\mathbf{x}_c}\right) + \lambda r\frac{\partial\mathbf{w}}{\partial\mathbf{x}_c} \tag{7}$$

where $r = \frac{\partial\Omega}{\partial\mathbf{w}}$, where $\Omega = \|\cdot\|_1$ for LASSO. Note that, for LASSO $r = \text{sub}(\mathbf{w})$, where $\text{sub}(\mathbf{w})$ is the sub-gradient of l_1 norm. The subgradient is defined as $+1$ is for each positive element of $\mathbf{w}$, -1 for each negative element and zero for elements that are zero.

In Algorithm 1, poisoning ratio, $\mathcal{P}$, is an attack parameter given by the adversary that determines number of attack samples, q, to poison dataset. For each attack point $\mathbf{x}_c$, Algorithm 1 first learns f by iteratively updating w and b as shown in Eq. (1). The attack algorithm uses Eq. (1) to compute the adversary's objective in Eq. (6). A projection operator $\Pi_{\mathcal{B}}(\mathbf{x})$ is used to project data point $\mathbf{x}_c$ onto a feasible box-constrained domain $\mathcal{B}$. This projection helps define gradient direction $\mathbf{d}$ of the attack point onto the constrained domain $\mathcal{B}$ and calculate gradient step size η by performing a line search. It is important to note that any arbitrary boundary set by the attacker will not generate concrete attacks. Boundary plays a critical role in both the effectiveness and detectability of an attack. A large choice for projection operator, $\Pi_{\mathcal{B}}(\mathbf{x})$, gives higher attack strength but is easily detectable [38]. Therefore, in these experiments we fix the bounding box to 0.5 that makes the poison sample more difficult to distinguish and, hence, less detectable.

VI. EXPERIMENTS AND RESULTS

In this section, we present an empirical analysis on the feasibility of embedded FS algorithms used in [13] against

information-theoretic methods. The experiments include several synthetic and real-world datasets. Our objective with these experiments is to answer the following questions:

1) Are information-theoretic FS algorithms robust against direct gradient attacks, such as those developed by Xiao *et al.* [13]?
2) Even though LASSO attack, a gradient attack, is not designed to directly attack information-theoretic FS algorithms, do these attacks still negatively impact the performance of the filter FS algorithms (e.g., consistency and adversarial similarity)?
3) Are certain information-theoretic FS algorithms impacted more than others by the poisoning attacks?

In addition to the information-theoretic FS methods discussed in Section IV-B, we also implemented Relief [39], and the Fisher score [40]. Note that FS using Relief and Fisher scores are not based on information-theoretic quantities; however, we feel it is also important to understand the transferability of Xiao *et al.*'s attack on FS methods that are also independent from LASSO. The experiments were implemented using Python and the scikit-feature library [41]. Scikit-feature implements the information-theoretic estimators. The reproducible code and data to for the experiments presented in this work be made publicly available on Github after publication.

A. DATASETS

The datasets used to benchmark the FS models discussed in this work were collected from the UCI machine learning database [42] and preprocessed following the standardization approach by Fernández-Delgado *et al.* [43]. Table 1 shows the properties of the datasets used. The datasets selected for the benchmarks are all binary classification tasks. We limit the experiments to binary prediction problems for several reasons. First, the original experiments with Xiao's LASSO attack algorithm focused on binary classification and that is the approach in the formulation of their gradient ascent algorithm. Second, mutli-class FS problems add a degree of freedom to the experiments that would limit our ability to accurately answer the questions posed above. Therefore, we evaluate binary classification datasets. Each dataset has an 80:20 split for training and testing data, respectively. We generated a maximum of 20% malicious samples for each training dataset, which were later injected in different proportions with benign training data D_{tr}, to generate malicious dataset D^*_{tr}. For the task of FS, information-theoretic algorithms select the top 30% of the total features from each dataset (see Table 1). All the results were generated and averaged over 15 cross-validation sets.

B. FIGURES OF MERIT

There are several figures of merit that are of interest when we measure the performance of a FS algorithm on a dataset. The first figure of merit is based on the similarity between the features selected by FS algorithms with and without and adversary when the FS model is based on information theory. One may ask, *What does "adversarial similarity" mean?* This question refers to the difference in features selected from poisoned data and the features selected from benign data. This figure of merit based on the similarity between benign and adversarial feature sets helps us understand the change in selection patterns with the addition of adversarial data. The second figure of merit is based on consistency of the information-theoretic FS algorithms. Consistency refers to the repeatability of selection of top p features [25], [26], [44]. For both of these figures of merit, we select the Kuncheva and Jaccard indices as figure of merit for similarity and consistency [25], [45].

The Kuncheva and Jaccard indices for similarity and consistency are useful for measuring the performance in terms of the stability of the feature selector as well as the impact that the adversary has on the features that are selected. Note that we can use the Jaccard index to compute the consistency and the similarity. The only difference is the input to the function that calculates the index. The same concept applies to the Kuncheva index as well. Further, it is important to see that the similarity and consistency capture information that cannot be measured by metrics such as classification error. For example, consider two subsets of features $\mathcal{A}_1$ and $\mathcal{A}_2$ that are vastly different but have the same classification errors ϵ_1 and ϵ_2 with the same base classifier. In this hypothetical example, the classification errors show no difference; however, the feature sets would have a very low consistency because they are significantly different. The Kuncheva and Jaccard scores are calculated as follows:

Kuncheva's Index [25]: Let $\mathcal{A}$ and $\mathcal{B}$ be defined as feature subsets of top p features that were selected from P possible features, and $r = |\mathcal{A} \cap \mathcal{B}|$ then Kuncheva's index is given by:

$$\mathsf{S}_{\text{Kunch}}(\mathcal{A}, \mathcal{B}, P) = \frac{rP - p^2}{p(P - p)}$$

which has the range $[\pm 1]$.

Jaccard's Index [45]: Let $\mathcal{A}$ and $\mathcal{B}$ be defined as feature subsets of length p that were selected from P possible features then Jaccard's index is given by:

$$\mathsf{S}_{\text{Jacc}}(\mathcal{A}, \mathcal{B}) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|}$$

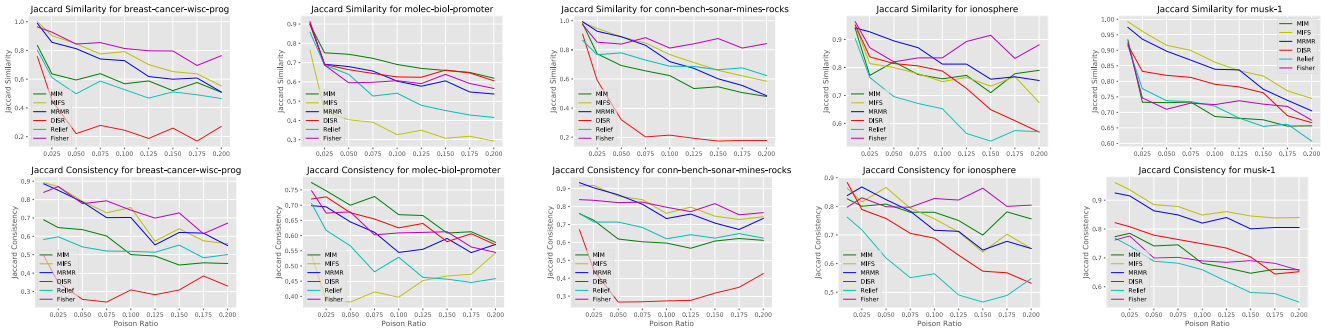
which has the range $[0, 1]$.

C. EXPERIMENTS

This section presents the results for several experiments that demonstrate the effect of data poisoned by LASSO attacks on FS tasks. The first set of experiments shows the impact of LASSO attacks on the similarity of features selected from benign and adversarial data. The second set of experiments show the effects of LASSO attacks on the consistency of FS algorithms. These experiments further explore the impact of the poisoning ratio (PR) on the two figures of merit. In all of the experiments, a vector of $\mathcal{P}$ has values that range from 1% to 20% with a step size of 2.5 is used to poison the training

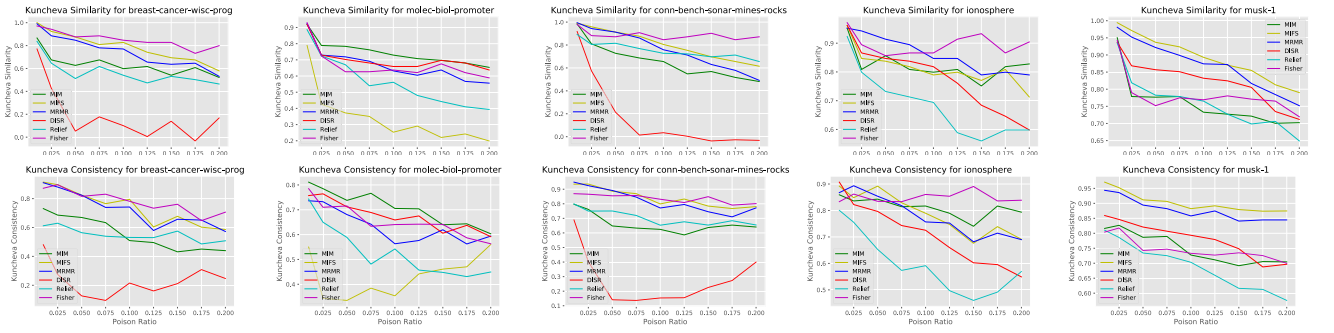
TABLE 1. Properties of the UCI Datasets used in the experimental benchmarks.

Datasets	Total features	Total samples	Training Data	Testing Data	Total Attack Samples	Features Selected
Breast-cancer Wisc. Prog.	33	198	158	40	31	10
Molecular Biology Promoter	58	106	84	22	16	18
Conn. Bench Sonar Mines	64	208	166	42	33	20
Ionosphere	35	351	281	70	56	11
Musk-1	166	476	381	95	76	50



(a) Breast Cancer (b) Molecular Biology (c) Connectionist Bench (d) Ionosphere (e) Musk-1

FIGURE 2. Similarity and Consistency scores for Jaccard on the benchmark datasets. *First row* shows the Jaccard Distance and *Second Row* shows the Jaccard Consistency delineating the performance of information-theoretic FS algorithms: MIM, MIFS, mRMR, DISR, Relief, and Fisher after the adversarial attack.



(a) Breast Cancer (b) Molecular Biology (c) Connectionist Bench (d) Ionosphere (e) Musk-1

FIGURE 3. Similarity and consistency scores for Kuncheva index on benchmark datasets. *First row* shows the Kuncheva Distance and *Second Row* shows the Kuncheva Consistency delineating the performance of the information-theoretic FS algorithms: MIM, MIFS, mRMR, DISR, Relief, and Fisher after the adversarial attack.

data (i.e., 1%, 2.5%, 5%, 7.5%, 10%, 12.5%, 15%, 17.5%, 20%). A bounding box value of 0.5 is used to generate attack data using Algorithm 1.

1) EFFECT OF POISONING PERCENTAGE ON INFORMATION-THEORETIC FEATURE SELECTION SIMILARITY

The similarity metric described in Section VI-B is a measure for features selected from the benign and adversarial dataset using information-theoretic FS algorithms. The natural interpretation of the similarity scores of Jaccard and Kuncheva is equal to one, if the feature sets from the poisoned and benign datasets are identical. As the Jaccard and Kuncheva scores drop, the interpretation is that the feature sets between the poisoned and benign datasets are becoming increasingly different. We use different poisoning ratios in the training data to study the effect on selection patterns by different information-theoretic FS techniques, namely MIM, MIFS, mRMR, and DISR and similarity based FS techniques, namely Relief and Fisher score.

Table 2 shows the Jaccard and Kuncheva similarity scores of features selected using filter-based FS algorithms from adversarial data on all benchmark datasets. The table shows the Jaccard/ Kuncheva scores for 1%, 5%, 10%, and 20% poisoning ratios. From the table, we observe that the FS algorithms are negatively impacted by the LASSO attack, which was originally designed for embedded FS approaches. Moreover, the impact of attack increases with an increase in poisoning ratios (i.e., Jaccard and Kuncheva similarity scores decrease with increasing poisoning ratio). This is an expected result since other works on data poisoning have shown that the adversaries' impact on the performances increases as their budget increases (i.e., how many samples can be

TABLE 2. Jaccard and Kuncheva similarity measure for Breast-Cancer, Molecular-Biology, Connectionist Bench, Ionosphere, and Musk-1 generated for various information-theoretic feature selection algorithms generated for poisoning ratio 1%, 5%, 10% and 20%.

Dataset	Poisoning Ratio	Jaccard Distance						Kuncheva Distance					
		MIM	MIFS	mRMR	DISR	Relief	Fisher	MIM	MIFS	mRMR	DISR	Relief	Fisher
Breast-cancer	1%	0.83	1.0	0.99	0.76	0.80	96	0.86	1.0	0.99	0.77	0.83	0.97
	5%	0.59	0.84	0.81	0.22	0.50	0.84	0.63	0.87	0.85	0.05	0.51	0.87
	10%	0.57	0.80	0.73	0.24	0.52	0.81	0.60	0.83	0.77	0.10	0.54	0.85
	20%	0.51	0.55	0.51	0.27	0.46	0.76	0.52	0.58	0.53	0.17	0.46	0.79
Molecular Biology	1%	0.90	0.76	0.90	0.89	0.86	0.91	0.91	0.79	0.92	0.92	0.88	0.93
	5%	0.74	0.40	0.68	0.66	0.64	0.60	0.78	0.37	0.72	0.70	0.70	0.63
	10%	0.70	0.32	0.60	0.63	0.54	0.60	0.73	0.25	0.63	0.66	0.56	0.64
	20%	0.62	0.30	0.53	0.60	0.41	0.56	0.65	0.20	0.55	0.63	0.39	0.59
Connectionist Bench	1%	0.98	0.99	0.99	0.91	0.86	0.97	0.98	0.99	0.99	0.92	0.892	0.98
	5%	0.69	0.89	0.89	0.32	0.78	0.84	0.73	0.91	0.91	0.21	0.81	0.87
	10%	0.62	0.77	0.72	0.21	0.69	0.81	0.65	0.80	0.76	0.03	0.73	0.84
	20%	0.48	0.58	0.48	0.18	0.62	0.84	0.48	0.61	0.49	-0.03	0.65	0.87
Ionosphere	1%	0.94	0.95	0.94	0.95	0.9	0.96	0.95	0.96	0.95	0.96	0.92	0.97
	5%	0.82	0.8	0.89	0.81	0.7	0.82	0.86	0.84	0.91	0.85	0.73	0.86
	10%	0.76	0.75	0.81	0.79	0.65	0.83	0.8	0.79	0.85	0.82	0.69	0.87
	20%	0.79	0.67	0.75	0.57	0.57	0.88	0.83	0.71	0.79	0.6	0.6	0.9
Musk-1	1%	0.93	0.99	0.97	0.92	0.91	0.92	0.95	0.99	0.98	0.94	0.94	0.94
	5%	0.73	0.92	0.9	0.82	0.74	0.71	0.78	0.94	0.92	0.86	0.78	0.75
	10%	0.69	0.86	0.84	0.79	0.72	0.72	0.73	0.89	0.87	0.83	0.77	0.77
	20%	0.66	0.74	0.7	0.67	0.61	0.67	0.7	0.79	0.75	0.71	0.65	0.72

TABLE 3. Jaccard and Kuncheva consistency measure for Breast-Cancer, Molecular-Biology, Connectionist Bench, Ionosphere, and Musk-1 generated for various information-theoretic feature selection algorithms generated for poisoning ratio 1%, 5%, 10% and 20%.

Dataset	Poisoning Ratio	Jaccard Consistency						Kuncheva Consistency					
		MIM	MIFS	mRMR	DISR	Relief	Fisher	MIM	MIFS	mRMR	DISR	Relief	Fisher
Breast-cancer	1%	0.69	0.89	0.89	0.49	0.58	0.84	0.73	0.91	0.91	0.48	0.61	0.87
	5%	0.64	0.79	0.79	0.26	0.54	0.78	0.67	0.82	0.82	0.13	0.56	0.82
	10%	0.5	0.76	0.7	0.31	0.52	0.74	0.51	0.8	0.74	0.22	0.53	0.78
	20%	0.45	0.56	0.55	0.33	0.5	0.67	0.44	0.59	0.57	0.25	0.51	0.71
Molecular Biology	1%	0.77	**0.54**	0.7	0.72	0.71	0.75	0.81	0.55	0.74	0.76	0.74	0.79
	5%	0.7	0.38	0.65	0.68	0.57	0.68	0.74	0.33	0.68	0.71	0.59	0.72
	10%	0.67	0.4	0.54	0.63	0.53	0.61	0.71	0.35	0.56	0.66	0.54	0.64
	20%	0.58	**0.54**	0.57	0.57	0.46	0.54	0.6	0.56	0.59	0.59	0.45	0.56
Connectionist Bench	1%	0.76	0.92	0.93	0.67	0.76	0.84	0.8	0.93	0.95	0.69	0.8	0.87
	5%	0.62	0.86	0.87	0.27	0.71	0.82	0.65	0.89	0.89	0.14	0.75	0.85
	10%	0.6	0.76	0.73	0.27	0.62	0.8	0.62	0.8	0.77	0.15	0.65	0.83
	20%	0.61	0.74	0.73	0.43	0.62	0.77	0.64	0.78	0.77	0.4	0.65	0.8
Ionosphere	1%	0.83	0.86	0.84	0.88	0.76	0.8	0.86	0.89	0.87	0.91	0.8	0.83
	5%	0.81	0.87	0.82	0.76	0.62	0.8	0.84	0.89	0.85	0.8	0.65	0.83
	10%	0.78	0.76	0.72	0.69	0.56	0.83	0.82	0.79	0.76	0.73	0.59	0.86
	20%	0.76	0.65	0.65	0.53	0.55	0.8	0.79	0.69	0.69	0.55	0.57	0.84
Musk-1	1%	0.77	0.96	0.92	0.82	0.77	0.76	0.82	0.97	0.94	0.86	0.81	0.8
	5%	0.74	0.88	0.86	0.78	0.69	0.7	0.79	0.91	0.89	0.82	0.73	0.74
	10%	0.68	0.85	0.82	0.75	0.66	0.69	0.73	0.88	0.86	0.79	0.7	0.73
	20%	0.66	0.84	0.81	0.65	0.55	0.66	0.7	0.87	0.84	0.7	0.58	0.7

added to the training data) [46], [47]. The results are shown in Figures 2 and 3 for Jaccard and Kuncheva scores, respectively. The first row of the two figures shows Jaccard and Kuncheva similarities for the algorithms on different levels of the poisoning ratio $\mathcal{P}$. All the algorithms are represented by different colored lines and show a declining trend on all datasets as the poisoning ratio increases. We also observe there is a sharp drop in the figure of merit until the poisoning ratio is 5% then declines gradually. It is interesting to note that similarity score of DISR goes as low as 0.2 for higher poisoning ratios from 0.9 for 1% poisoning. Fisher shows the smallest overall decline. Algorithms such as MIFS, mRMR, DISR shows a monotonic decline. Finally, these results show that even though the attacks were developed for LASSO, there is a negative impact on the FS performance.

2) EFFECT OF POISONING RATIO ON INFORMATION-THEORETIC FEATURE SELECTION CONSISTENCY

Consistency metric measures the repeatability of top p features when a FS algorithm is run on bootstrap samples from a dataset. The experiments are performed by generating a poisoned dataset (e.g., 5% adversarial and 95% benign) then running FS on fifteen bootstrap datasets of the same size from the poisoned dataset. These scores are independent of the benign dataset. Hence, this experiment is different from the first experiment using the similarity score. This experiment shows how consistent the feature set is in the presence of an adversary. It is important to understand that FS may result in a consistent feature set; however, the feature set is *consistently* different than the feature set if only benign data were used.

These experiments in this section were conducted similarly to the previous one that uses the same values of poisoning ratios and information-theoretic FS algorithms.

Table 3 shows Jaccard and Kuncheva consistency scores for the adversarial datasets with varying degrees of poisoning data. From the table, we observe that the Jaccard and Kuncheva consistency scores decrease as the poisoning ratio in D_{tr}^{*} increases. The results are shown in Figures 2 and 3. Second row of the two figures show the consistency plots. It is interesting to note that the consistency scores for FS algorithms like DISR increases for the breast cancer and connectionist-bench datasets and similarly in MIM for molecular biology dataset. However, there is an overall decrease in the Jaccard and Kuncheva consistencies from poisoning ratios between 1% to 20% for all algorithms on all datasets, except for MIFS in molecular biology where there is no net decrease in consistency score.

Further, we see from Figure 2 and 3 that the consistency drop is gradual and even increases for certain algorithms, as discussed previously. This can appear to have mild or non-harming effect of increasing poisoning ratios on consistency, if taken in isolation. However, if we observe the similarity and consistency score as couple, we observe that the effect of gradual decay of consistency is rather drastic since the corresponding similarity scores are very low. In other words, even though the repeatability of features is not too low, the features being selected are largely dissimilar from legitimate features. Thus, the features that are being selected are consistently wrong (i.e., they do not agree with the features that are selected from a benign dataset). It is, therefore, important to use similarity and consistency scores together.

VII. CONCLUSION

Feature selection and data preprocessing remains a vital step in nearly all data science and machine learning pipelines; however, many applications that rely on machine learning can easily be fooled by adversarial samples. Further, adversarial data are of increasing concern because of how easy the data are to generate. A considerable amount of work has been done on the impacts of the adversary on a classifier; however, the adversary's effects on feature selection are somewhat limited, even though the consequences of attacks on both classifier and feature-selectors are detrimental. Our work shows how adversarial attacks designed to poison embedded FS algorithms (e.g., LASSO) can be hostile to information-theoretic-based FS algorithms. Even though the two LASSO and information-theoretic approaches are structurally very different, the result is the same with the adversarial data: *performance is reduced*. This observation is the answers to Question #1 and #2 in Section VI. This work also shows the degree of damage poisoning attacks can have on information-theoretic FS algorithms by injecting a range of poisoning samples into the benign dataset. This study also shows how there could be a false sense of security when the consistency of a FS algorithm is relatively high, but the similarity between benign and adversarial features sets are low. Not only could there be a false sense of security but some FS algorithms are impact more by adversarial data than others. This observation addresses Question #3 in Section VI.

Our future work includes developing attacks that directly target information-theoretic FS algorithms and evaluate the transferability of correlation-based FS and Lasso. Further, developing FS based on information theory that incorporates robustness measures against adversaries is also of broader interest to the community based on the findings in this work.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

The authors would like to thank Eduardo Zambrana for help with the initial development of the code.

REFERENCES

[1] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar, "Can machine learning be secure?" in *Proc. ACM Symp. Inf., Comput. Commun. Secur. (ASIACCS)*, 2006, pp. 16–25.

[2] D. Maiorca, I. Corona, and G. Giacinto, "Looking at the bag is not enough to find the bomb: An evasion of structural methods for malicious PDF files detection," in *Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur. (ASIA CCS)*, 2013, pp. 119–130.

[3] D. Lowd and C. Meek, "Good word attacks on statistical spam filters," in *Proc. 2nd Conf. Email Anti-Spam*, 2005, pp. 1–8.

[4] D. Bekerman, B. Shapira, L. Rokach, and A. Bar, "Unknown malware detection using network traffic classification," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Sep. 2015, pp. 134–142.

[5] Y. Zhang and Z. Gan, "Generating text via adversarial training," in *Proc. Adv. Neural Inf. Process. Syst., Workshop Adversarial Training*, 2016, pp. 21–32.

[6] W. Xu, Y. Qi, and D. Evans, "Automatically evading classifiers: A case study on PDF malware classifiers," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2016.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–11.

[8] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," 2017, *arXiv:1702.02284*. [Online]. Available: http://arxiv.org/abs/1702.02284

[9] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1467–1474.

[10] H. Xiao, H. Xiao, and C. Eckert, "Adversarial label flips attack on support vector machines," in *Proc. Eur. Conf. Artif. Intell.*, 2012, pp. 870–875.

[11] G. Ditzler and A. Prater, "Fine tuning lasso in an adversarial environment against gradient attacks," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–7.

[12] K. K. Budhraja and T. Oates, "Adversarial feature selection," in *Proc. IEEE Int. Conf. Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 288–294.

[13] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1689–1698.

[14] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[15] G. Brown, A. Pocock, M.-J. Zhao, and M. Lujan, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, pp. 27–66, Jun. 2012.

[16] K. Sadeghi, A. Banerjee, and S. K. S. Gupta, "A system-driven taxonomy of attacks and defenses in adversarial machine learning," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 4, pp. 450–467, Aug. 2020.

[17] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *Proc. IEEE Conf. Commun. Netw. Secur.*, Oct. 2014, pp. 247–255.

[18] G. Qu, S. Hariri, and M. Yousif, "A new dependency and correlation analysis for features," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 9, pp. 1199–1207, Sep. 2005.
[19] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. Milton Park, U.K.: Taylor and Francis Group, 2011.
[20] C.-T. Lin, N.-J. Wang, H. Xiao, and C. Eckert, "Feature selection and extraction for malware classification," *J. Inf. Sci. Eng.*, vol. 31, pp. 965–992, 2015.
[21] U. Pehlivan, N. Baltaci, C. Acarturk, and N. Baykal, "The analysis of feature selection methods and classification algorithms in permission based Android malware detection," in *Proc. IEEE Symp. Comput. Intell. Cyber Secur. (CICS)*, Dec. 2014, pp. 1–8.
[22] S. Hess, P. Satam, G. Ditzler, and S. Hariri, "Malicious HTML file prediction: A detection and classification perspective with noisy data," in *Proc. IEEE/ACS 15th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2018, pp. .
[23] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B, Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.
[24] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," 2017, *arXiv:1712.03141*. [Online]. Available: http://arxiv.org/abs/1712.03141
[25] L. I. Kuncheva, "A stability index for feature selection," in *Proc. Int. Conf. Artif. Intell. Appl.*, 2007, pp. 390–395.
[26] S. Nogueira, K. Sechidis, and G. Brown, "On the stability of feature selection algorithms," *J. Mach. Learn. Res.*, vol. 19, pp. 1–54, 2018.
[27] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, nos. 1–4, pp. 131–156, 1997.
[28] P. E. Meyer, C. Schretter, and G. Bontempi, "Information-theoretic feature selection in microarray data using variable complementarity," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 3, pp. 261–274, Jun. 2008.
[29] B. Senliol, G. Gulgezen, L. Yu, and Z. Cataltepe, "Fast correlation based filter (FCBF) with a different search strategy," in *Proc. 23rd Int. Symp. Comput. Inf. Sci.*, Oct. 2008, pp. 1–4.
[30] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the NIPS 2003 feature selection challenge," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1–8.
[31] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2006.
[32] D. D. Lewis, "Feature selection and feature extraction for text categorization," in *Proc. Workshop Speech Natural Lang. (HLT)*, 1992, pp. 212–217.
[33] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[34] H. Liu and G. Ditzler, "Speeding up joint mutual information feature selection with an optimization heuristic," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2017, pp. 1–8.
[35] H. Liu and G. Ditzler, "A semi-parallel framework for greedy information-theoretic feature selection," *Inf. Sci.*, vol. 492, pp. 13–28, Aug. 2019.
[36] L. Huang, A. Joseph, B. Nelson, B. Rubinstein, and J. D. Tygar, "Adversarial maching learning," in *Proc. ACM Worskshop Artif. Intell. Secur.*, 2011, pp. 43–58.
[37] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statist. Soc., B, Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, 2005.
[38] C. Frederickson, M. Moore, G. Dawson, and R. Polikar, "Attack strength vs. detectability dilemma in adversarial machine learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.
[39] K. Kira and L. Rendell, "A practical approach to feature selection," in *Proc. Nat. Conf. Artif. Intell.*, 1992, pp. 249–256.
[40] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2001.
[41] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, Jan. 2018.
[42] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[43] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.
[44] S. Nogueira and G. Brown, "Measuring the stability of feature selection," in *Proc. Eur. Conf. Mach. Learn.*, 2016, pp. 442–457.
[45] P. Français, *Bulletin de la Société Vaudoise des Sciences Naturelles*, vol. 37. pp. 547–579.
[46] H. Liu and G. Ditzler, "Poisoning MRMR with adversarial data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2019, pp. 2517–2521.
[47] H. Liu and G. Ditzler, "Data poisoning against information-theoretic feature selection," *Inf. Sci.*, vol. 573, pp. 396–411, Sep. 2021.

SRISHTI GUPTA received the B.Tech. degree in electronics and communication engineering, India, in 2017. She is currently pursuing the M.S. degree with the Department of Electrical and Computer Engineering, The University of Arizona. Her research interests include data mining, feature selection, adversarial machine learning, and their applications in cybersecurity and fraud detection.

ROMAN GOLOTA is currently pursuing the bachelor's degree in electrical and computer engineering and mathematics with The University of Arizona. His research interest includes adversarial machine learning.

GREGORY DITZLER (Senior Member, IEEE) received the B.Sc. degree from the Pennsylvania College of Technology, in 2008, the M.Sc. degree from Rowan University, in 2011, and the Ph.D. degree from Drexel University, in 2015. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, The University of Arizona. His research interests include machine learning, adversarial learning, neural networks, concept drift, and applications life sciences, and cybersecurity. From 2016 to 2018, he was selected as a Summer Faculty Fellow with the Air Force Research Laboratory to work on adversarial learning algorithms. He received an Outstanding Article Award from *IEEE Computational Intelligence Society Magazine,* in 2018, the Best Paper at the IEEE International Conference on Cloud and Autonomic Computing, in 2017, and the Best Student Paper at the IEEE/INNS International Joint Conference on Neural Networks, in 2014. He was a recipient of the NSF CAREER Award. He is an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and *Cluster Computing*.