Attack on Distance Predictor by Mimicking Target's Optical Sample Distribution

Alex Ferrando Víctor Novelle Marc Ruiz Luis Velasco
Polytechnic University of Catalonia
{alex.ferrando,victor.novelle}@estudiantat.upc.edu {marc.ruiz-ramirez,luis.velasco}@upc.edu

Abstract

Machine Learning (ML) and Deep Neural Networks (DNNs) have been widely used in a large extent of applications, including several safety-critical ones, such as security mechanisms. As in recent years, it has been shown that these methods present vulnerabilities that can be exploited, great concern has been raised about the robustness of the current ML-based defenses. In order to ensure that a specific ML security system is robust against any possible threat, specific targeted attacks that aim to bypass the defense mechanism must be constructed, to later correct the found vulnerabilities.

In this paper, three attack methodologies against a ML-based data-filtering security system used on optical connections based on emitter's distance prediction are presented. These threats effectively allow the introduction of undesired data on the optical connection, fooling the proposed defense mechanism by mimicking the desired optical samples distributions. The results presented in this paper help to identify the vulnerability issues of the proposed security system so that research can be carried out on how to improve it.

1. Introduction

Since its invention in 1977, the purity of optical fibers has been refined for extremely efficient information transmission. This technology represents nowadays the backbone of fast communication networks that go over the entire globe. Along with the expansion of fiber optic facilities, there has been a growing concern to ensure that the communications that occur through them are secure [11, 12].

During the decade of 90s, several studies were carried out about how optical networks could be attacked as well as how to create mechanisms to protect the networks against them [16, 17]. In these researches, the proposed detection methods were mainly based on the establishment of thresholds on different signal magnitudes, such as the amplitude or the phase to identify eavesdropping or malicious distortions that could provoke a degradation in the communications.

In recent years, several studies have been carried out on the usage of ML techniques to improve defensive systems in optical communications [7,8,18]. Notice that these systems do not only aim to avoid denial of service (DoS) but to also be able to detect eavesdropping or the insertion of external signals into the communication.

The goal of the work developed and exposed throughout this paper is to formulate and test the performance of three different attack strategies that aim to bypass a ML defense system based on distance-to-sender filtering (presented in Section 3). These methods modify the distribution of the optical signals, so that data sent by the attacker from a closer distance to the receiver mimics the distribution of the original, being identified as correct by the defensive system. In this way, the victim would think that a manipulated/false message was sent from the trusted source while instead it could be sent by the attacker somewhere in the middle of the communication channel.

2. Related Work

Generally, the attacks performed over ML systems are classified into three different groups:

Poisoning attacks The aim of poisoning attacks consists in degrading the model performance through the insertion of specifically modified data. Thus, to perform this attack, the attacker is only supposed to have access to the training data pipeline of the victim ML system. Nevertheless, the performance of most of the prior work done in this area [13, 14,21,23] relies on the capability of being able to mimic the structure of the victim model.

Using $\mathcal{D}_{train}$ to denote the clean training data, $\mathcal{D}_{val}$ a validation data set, θ for the model parameters and being $\mathcal{P}(\cdot)$ the poisoning procedure, the spiteful objective can be formulated as:

$$\max_{\mathcal{P}} \mathcal{L}(\mathcal{D}_{val}; \theta^*) : \theta^* \in \arg\min_{\theta} \mathcal{L}(\mathcal{P}(\mathcal{D}_{train}); \theta) \quad (1)$$

where $\mathcal{L}$ represents the loss function used in the attacker's mock-up model.

Backdoor attacks In the threat of backdoor attacks, the attacking goal consists in inducing a mislead in the model for certain inputs of interest, without compromising the model accuracy on clean data. This is generally achieved with the injection of trigger patterns on the desired samples [1, 2, 25, 26]. Unlike the previous attack, in this case, the attacker must also have access to the inference data, to add the trigger.

The formulation of these attacks is equal to the poisoning case, but $\mathcal{D}_{val}$ in Eq.(1) is conformed by data samples that contain the trigger pattern instead of being a subset of the underlying distribution of $\mathcal{D}_{train}$.

Nevertheless, this attack system is not suitable for the attack we want to implement in this paper. The main reason is the lack of control over what is inputted to the network or what is not. We don't know which features will the network be using. In addition, due to the aggregation process carried out in the GMM, it remains impossible to settle exact values for the data inputted to the network.

Adversarial examples attacks In recent years, with the rise of deep learning, extensive research around the vulnerabilities of the networks has been performed, showing that adversarial examples attacks targeted to conventional ML [15, 22], that were studied theoretically decades ago, could now be implemented exploiting the weaknesses of DNNs [5].

In this type of attack, hardly perceptible perturbations by humans are introduced into images to generate a mislead in network-based classifiers. Even though multiple craft procedures exist for the perturbation generation [9, 10], the most widely used procedure [19, 20], which generally only assumes access to the test data by the attacker, consists in optimizing the following objective function:

$$\max_{x'} \mathcal{L}(x', y; \theta) : \|x' - x\|_p \leq \epsilon \tag{2}$$

where ϵ represents the maximum perturbation size in $\mathcal{L}^p$-norm, x the original test sample and x' the perturbed one.

3. Network attack

In [6], the authors presented a methodology to model the propagation of optical constellations through an optical system based on Neural Networks (NNs). In particular, the NN models are used to characterize optical components (fiber links and optical nodes) by means of selected Gaussian-based features and concatenated in order to emulate the propagation of a signal from the transmitter to the receiver, which allows obtaining in a fast and accurate way the optical constellations at intermediate and destination nodes. In addition, NN-based analysis can be also performed by analyzing monitored constellations at the receiver and estimating its characteristics, such as distance and launch power [24]. Thus, actual lightpath distance can be estimated by means of a NN model at the receiver and compared with the expected one; in case of a significant difference, an unexpected situation is detected.

One possible reason behind such an unexpected case is a tampering attack, where an attacker intercepts the original signal at an intermediate point of the path, performs a given attack (e.g. eavesdropping), and re-inserts the signal to avoid lightpath disruption and behave as a normal situation. Since the signal is now generated from a distance to the receiver different (shorter) from the expected one, the NN-based distance analysis model can easily detect tampering.

In our attack scenario, communication is assumed between two users of the network (sender and receiver) and an attacker who is able to hack into the communication between the users. The attacker in turn is able to intercept the messages sent by the sender and to send his messages to the receiver through the network. On the other hand, the receiver does not have any type of control over the network and is only able to discern if the message was by an attacker or by the secure sender through the analysis of the received symbols. As suggested above, the usage of Gaussian Mixture Modeling (GMM) for optical samples feature extraction and a NN distance predictor represents a way for the receiver to check whether the message was sent from the trusted source or an attacker in some mid-point of the network. A way to do that is to accumulate several symbols forming an optical sample, extracting the mean and variances of the symbols and inputting the values to a neural network to predict the distance from which they were sent. If the predicted distance is different from the expected one, we can suspect that a tampering attack is being produced and thus, the observation is discarded (see Figure 1).

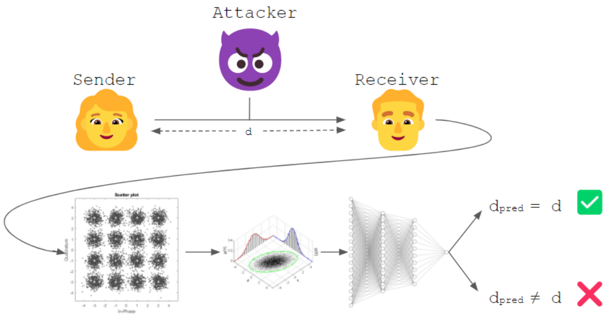

Figure 1. Defense system

Notice that once the optical connection has been compromised, the security breach generated is extremely large, being able to perform other types of attacks on systems downstream of the defensive method from the new malicious connection. The attacker's possibilities are extremely broad, being able to perform poisoning or backdoor threats with the introduction of malicious data that will be detected as original, in addition to being able to consult private information.

In the following subsections, the notation used throughout the paper is presented as well as a brief description of the considered threat models depending on the attacker's capabilities is introduced. Then, we present our proposed approaches for effective tampering attacks on the optical transmission setup introduced previously. A thorough evaluation of the performance of the suggested attacks is given in Section 4.

3.1. Notation

It is important to define the notation to understand the equations and formulas in the paper. We encourage the reader to pay attention to this paragraph since the presence of multiple indices in the variables might make the reading of the equations complex if the notation is not well understood.

We define a given optical sample sent from a distance d as a set of 16 constellation points $O^{(d)} = \{C_{(1)}^{(d)}, ..., C_{(16)}^{(d)}\}$. At the same time, each constellation point is defined as a set of n symbols x placed in the two-dimensional complex plane, so: $C_{(i)}^{(d)} = \{x_{(i),1}^{(d)}, ..., x_{(i),n}^{(d)}\}$. In this way, given a symbol from an optical sample (*e.g.* $x_{(i),j}^{(d)} = 3+2i$), we can easily identify that it has been sent from a distance d and received in the i-th constellation point. Therefore, the set of points received in each constellation point can be considered as generated from a 2-dimensional unknown probability distribution $x_{(i),j}^{(d)} \sim \mathbf{P}(d,i)$ depending on the distance from which the symbol was sent and the constellation point to which it belongs to. The distribution's first and second-order moments are indicated respectively as $\mu_{(i)}^{(d)}$ and $\Sigma_{(i)}^{(d)}$. In addition, subindices will be added to the moments to indicate the inner elements from the mean vector and the covariance matrix. In the attack process, we will want to mimic a distribution $\mathbf{P}(d^*, i)$ where d^* represents the target distance. Last, we will use an apostrophe to indicate that a result is raised from the manipulated symbols (*e.g.* $x'^{(d)}_{(i),j}$ would represent a manipulated symbol and $\mu'^{(d)}_{(i)}$ the mean of the manipulated symbols sent from distance d in the i-th constellation point).

3.2. Threat model

Two main threat scenarios have been envisioned against the proposed defensive system, which differ in the degree of knowledge of the target system by the attacker and the capability of manipulating the input data of any component in the network. The use of different attacker assumptions allows defining different attack strategies with a common goal, achieving a security breach on the victim system in multiple ways. [3, 15]

Advanced Knowledge Scenario (AK) In this scenario, it is assumed that the attacker knows everything about the system structure except for the neural network (number of layers, parameters, loss function used ...), which remains a black-box as well as how the GMM has been implemented. The attacker has access to both the optical samples $O^{(d)}$ and the extracted features from the Gaussian Mixture Model pre-processing $\{\mu_{(i)}^{(d)}, \Sigma_{(i)}^{(d)}\}$. Although in this setting the attacker has excessive capabilities that rarely will be available in a real attack, it enables us to perform an in-depth evaluation of the security of the defensive system.

Limited Knowledge Scenario (LK) In this setup, the attacker has only access to the raw optical sample data $O^{(d)}$ and has only partial awareness of the security system's structure of the victim. More concretely, the attacker knows the existence of a pre-processing phase between the optical sample transmission and the distance-to-sender computation that extracts the first and second-order moment of the symbols, but neither knows which pre-processing technique is used nor which kind of model is applied for the distance estimation. Even though both components remain as a black-box for the attacker, we will assume that it is apprised of the fact that the victim computations are based on using the first and second-order moments from an unknown probability distribution.

Thus, this scenario represents a more strict grey-box setting than the previous case but it mimics better the situation that an attacker will be facing when executing a real threat.

3.3. Feature mimic attack

In the AK scenario, the capability of modifying the data in any stage of the optical transmission will be exploited to perform a direct and effective attack. The fundamental insight behind our proposed attack for this threat model is to modify the output features of the GMM pre-processing instead of the original optical samples. By doing this, the attacker can avoid a dilution effect in the modified data, as it will be directly introduced as input to the network instead of having to be processed by the GMM component, which may imply a weaker performance.

Our suggested attack tries to mimic the values of the first and second-order moments obtained from the attacker's available data (D_{Att}) of each constellation point depending on the distance $\mathbf{d}$ using polynomial regression $\mathcal{P}$. In this way, if our approximation is good enough we will be able to mimic distances for which we have no data. Thus, the objective of this attack can be formulated as an optimization problem for each component of each extracted feature:

$$\min_{\mathcal{P}_{\mu_{(i),j}}} \left|\mu^{(\mathbf{d})}_{(i),j} - \mathcal{P}_{\mu_{(i),j}}(\mathbf{d})\right|^2 \quad j \in [1,2]$$
$$\min_{\mathcal{P}_{\Sigma_{(i),jk}}} \left|\Sigma^{(\mathbf{d})}_{(i),jk} - \mathcal{P}_{\Sigma_{(i),jk}}(\mathbf{d})\right|^2 \quad j,k \in [1,2] \tag{3}$$
$$\forall \mathbf{d}, i$$

After solving the presented equations, a polynomial for each feature (means and covariances) for each constellation point is obtained. Then, they are used to modify the original extracted features using the following formula:

$$\mu'^{(d)}_{(i),j} = \mu^{(d)}_{(i),j} + \mathcal{P}_{\mu_{(i),j}}(d^*) - \mathcal{P}_{\mu_{(i),j}}(d)$$
$$\Sigma'^{(d)}_{(i),jk} = \Sigma^{(d)}_{(i),jk} + \mathcal{P}_{\Sigma_{(i),jk}}(d^*) - \mathcal{P}_{\Sigma_{(i),jk}}(d) \tag{4}$$
$$\forall i$$

Notice that the modified features obtained using Eq.(4) will have the same variance than the original ones as we are adding the difference between the values of the polynomial approximation in the real and target distances. This event does not represent an inconvenience, since the variance of the parameters remains constant or increases with distance. Thus, thanks to this phenomenon, our modified features variance will stay in the range expected by the network, ensuring a successful attack.

3.4. Symbol-to-symbol modification attack

Once we have analyzed the problem considering that we can directly modify the features extracted from the optical samples by the GMM, the Limited Knowledge Scenario is further analyzed. In this setup, the victim of the attack would receive a modified optical sample $C'^{(d)}$ and would extract the features needed to feed the neural network from it.

Nevertheless, to guarantee that the modifications applied on the raw symbols conform to those that could occur in a real practical scenario, the set of possible transformations on 2-dimensional input symbols has been limited to the affine ones. Therefore, the ultimate goal of this attack will be to find a matrix $M^{(d \to d^*)}_{(i)}$ and a vector $b^{(d \to d^*)}_{(i)}$ for each constellation point i so that we end up with a set of transformed symbols $C'^{(d)}_{(i)} = \{x'^{(d)}_{(i)}\}_j$ obtained by:

$$x'^{(d)}_{(i),j} = M^{(d \to d^*)}_{(i)} x^{(d)}_{(i),j} + b^{(d \to d^*)}_{(i)} \tag{5}$$

that resemble the extracted features from $D^{(d)}_{Att}$ in the preprocessing stage in the i-th constellation point in the target distance d^*. In other words, we want the first and second order moments of the modified constellation point to be similar to the those of the i-th constellation point in the target distance d^*.

$$\mathbb{E}[C'^{(d)}_{(i)}] = \mu'^{(d)}_{(i)} \approx \mu^{(d^*)}_{(i)}$$
$$Cov(C'^{(d)}_{(i)}) = \Sigma'^{(d)}_{(i)} \approx \Sigma^{(d^*)}_{(i)} \tag{6}$$

Note that if the attacker does not have data from the target distance, the values of the first and second moments can be interpolated using the data from the other available distances in $D^{(d)}_{Att}$ and the process explained in the first paragraph of section 3.3.

Z-score attack A fast and direct implementation to transform the symbols is to normalize the distribution of the points we want to modify in the real distance and denormalize it with the computed covariance and mean of the symbols' distribution in the same constellation point in the target distance. Since $\Sigma^{(d)}_{(i)}$ is a covariance matrix (i.e symmetric positive definite), there must be a decomposition $\Sigma^{(d)}_{(i)} = P^{(d)}_{(i)} \Lambda^{(d)}_{(i)} P^{T(d)}_{(i)}$, where $\Lambda^{(d)}_{(i)}$ is diagonal. In consequence, we can compute the modified symbols as:

$$z_1 = P^{(d)}_{(i)} \Lambda^{-\frac{1}{2}(d)}_{(i)} P^{T(d)}_{(i)} \left(x^{(d)}_{(i),j} - \mu^{(d)}_{(i)}\right)$$
$$x'^{(d)}_{(i),j} = P^{(d^*)}_{(i)} \Lambda^{\frac{1}{2}(d^*)}_{(i)} P^{T(d^*)}_{(i)} z_1 + \mu^{(d^*)}_{(i)} \tag{7}$$

being $M^{(d \to d^*)}_{(i)} = P^{(d^*)}_{(i)} \Lambda^{\frac{1}{2}(d^*)}_{(i)} P^{T(d^*)}_{(i)} P^{(d)}_{(i)} \Lambda^{-\frac{1}{2}(d)}_{(i)} P^{T(d)}_{(i)}$ the transformation matrix and the transformation vector $b^{(d \to d^*)}_{(i)} = \mu^{(d^*)}_{(i)} - P^{(d^*)}_{(i)} \Lambda^{\frac{1}{2}(d^*)}_{(i)} P^{T(d^*)}_{(i)} P^{(d)}_{(i)} \Lambda^{-\frac{1}{2}(d)}_{(i)} P^{T(d)}_{(i)} \mu^{(d)}_{(i)}$. It must be noted that, as we are doing an affine transformation, the resulting modified constellation point will have the same mean and covariance than the computed mean and covariance in the target distribution.

Gradient descent attack In the previous approach, a fundamental assumption is made: the means and covariances extracted from each constellation point compose the input to the NN. Even though this fact is true, according to the LK scenario definition, it is clear that this statement is highly motivated due to the structure of the proposed security system. Thus, if other security measures are used in the defense that do not only depend on the first and second-order moments of the constellation points' distribution, our proposed Z-Score attack would not work as expected.

Therefore, a more generalized procedure would be useful in the supposed case that our current security method is upgraded or modified. This method should adapt the affine transformation coefficients depending on the set of features used to perform the distance predictions. Our proposed method for this generalized set-up consists of the usage of the Gradient Descent algorithm. Let's $f_j : \mathbb{C}^n \to \mathbb{C}^k$ a function that extracts a k-dimensional feature from a set of n symbols and be the vector indicating an importance weight for each feature, we can formulate a minimum square loss function as:

$$\mathcal{L}^{GD}\left(M_{(i)}^{(d\to d^*)}, b_{(i)}^{(d\to d^*)}, \boldsymbol{\alpha}; C_{(i)}^{(d)}\right) = \sum_{j} \alpha_j \left(f_j(C'^{(d)}_{(i)}) - f_j(C^{(d^*)}_{(i)})\right)^2 \tag{8}$$

We could formulate the presented loss function for the specific set-up we try to attack as in Equation (9). In this case, the applied functions f_j are the ones that compute the covariance matrix and the mean vector from a set of symbols. So, the loss function for our scenario would be the one that measures the square difference between the elements from the covariance matrices and the means of the transformed symbols and the target symbols.

$$\begin{gathered}\mathcal{L}^{GD}(M_{(i)}^{(d\to d^*)}, b_{(i)}^{(d\to d^*)}, \alpha, \beta; \Sigma_{(i)}^{(d^*)}, \mu_{(i)}^{(d^*)}, \gamma) = \\ \alpha \sum_{j=1}^{2}\sum_{k=1}^{2}\left(\gamma\Sigma'^{(d)}_{(i),j,k} - \gamma\Sigma^{(d^*)}_{(i),j,k}\right)^2 + \beta\sum_{j=1}^{2}\left(\mu'^{(d)}_{(i),j} - \mu^{(d^*)}_{(i),j}\right)^2 \\ with \quad \Sigma'^{(d)}_{(i)} = M_{(i)}^{(d\to d^*)}\Sigma_{(i)}^{(d)}M_{(i)}^{T(d\to d^*)} \\ \mu'^{(d)}_{(i)} = M_{(i)}^{(d\to d^*)}\mu_{(i)}^{(d)} + b_{(i)}^{(d\to d^*)}\end{gathered} \tag{9}$$

There are some details of the equation that should be noted. The first one is the presence of two additional variables α and β that are used to penalize more or less the errors in the covariance matrix or the mean vector. The second one is that it is not needed to compute the mean and the covariance of the points in each step of the algorithm (which would be extremely expensive in time). Instead, as we are applying an affine transformation to the points, we can rapidly compute the resulting covariance matrix and mean vector without the necessity of transforming all the symbols at each step. Last, a γ value is added to deal with the bias of the variance. We must consider that if we are computing the covariance matrix dividing the sum of square differences by N instead of $N-1$, a bias indirectly proportional to the number of symbols is introduced. Thereby, if we are computing the gradient descent with an amount of symbols N and we intend to transform a set of symbols of size n, we will need $\gamma = \frac{N(n-1)}{(N-1)n}$.

The minimization of the loss function is done in an iterative way using the Gradient Descent algorithm [4]:

$$\begin{aligned} M_{(i)}^{(d\to d^*)t+1} &= M_{(i)}^{(d\to d^*)t} - \eta\nabla_{M_{(i)}^{(d\to d^*)t}}\mathcal{L}^{GD} \\ b_{(i)}^{(d\to d^*)t+1} &= b_{(i)}^{(d\to d^*)t} - \eta\nabla_{b_{(i)}^{(d\to d^*)t}}\mathcal{L}^{GD}\end{aligned} \tag{10}$$

4. Experiments

4.1. Basic Settings

Dataset We conduct our experiments using an optical sample dataset provided by the Advanced Broadband Communications Center (*CCABA*) research center at the Polytechnic University of Catalonia (*UPC*). This dataset contains 50 observations of optical samples, each one of them containing 2048 symbols, for 25 different distances, starting at 80 km and increasing with a step of the same distance until a maximum distance of 2000 km.

Evaluation metric Our attacks evaluation is mainly based on forwarding the modified data through the NN and obtaining the loss of the network in inference time. Notice that in the AK setup, the modified data is directly introduced as input whereas, in LK, the GMM step is executed previously to the network forwarding phase. Our objective is to obtain a network loss that is similar to the one present in the NN when using real observations. Thus, to evaluate such similarity, the following value is proposed as an evaluation metric:

$$\begin{gathered}\kappa^{(d\to d^*)} = |\mathcal{L}^{NN}(\{\boldsymbol{\mu'^{(d)}}, \boldsymbol{\Sigma'^{(d)}}\}, d^*;\theta) - \\ \mathcal{L}^{NN}(\{\boldsymbol{\mu^{(d^*)}}, \boldsymbol{\Sigma^{(d^*)}}\}, d^*;\theta)| \\ with \quad \mathcal{L}^{NN}(\{\boldsymbol{\mu^{(d)}}, \boldsymbol{\Sigma^{(d)}}\}, d^*;\theta) = \\ \frac{1}{n}\sum_{i=1}^{n}\left|d^* - f(\{\boldsymbol{\mu^{(d)}}, \boldsymbol{\Sigma^{(d)}}\}_i;\theta)\right| \quad | \quad f(\cdot;\theta): \mathbb{R}^{24} \to \mathbb{R}\end{gathered} \tag{11}$$

being $f(\cdot;\theta)$ the NN, θ the network parameters, $\{\boldsymbol{\mu^{(d^*)}}, \boldsymbol{\Sigma^{(d^*)}}\}$ the real set of features for distance d^* and $\{\boldsymbol{\mu'^{(d)}}, \boldsymbol{\Sigma'^{(d)}}\}$ the set of features of the modified optical sample.

Network architecture and training We run our experiments using as distance predictor a small fully-connected neural network composed of three layers. This model is trained for 5000 epochs employing the RMSprop optimizer with learning rate 1×10^{-4} and ρ (discounting factor) equal to 0.9.

Indicate that in all the performed experiments, 10 independent runs with random splits for both the neural network training data and the D_{Att} selection have been executed, averaging the results afterward, to obtain a more robust metric.

4.2. Feature mimic attack performance

In this subsection we present the experiments that have been conducted, using the set-up described above, to see how our proposed approach for the Advanced Knowledge scenario performs under different conditions. Mainly, there are two factors that can greatly impact the performance of the attack.

On the one hand, the number of distances that the attacker has access to (β) and how they are distributed. Even though in the AK setup the attacker can theoretically obtain all the features of the desired distance, it is interesting to visualize how the polynomial approximation can fail to mimic the features' distributions if only some of them are available. In the conducted experiments, $\beta \in \{25, 12, 6\}$) have been used, selecting them equidistantly within the range of available distances.

On the other hand, the decision of which order is used to perform the polynomial approximation is a crucial step in this attack. A high order implies a better mimic of the distribution, which is desired when having enough data from all the possible distances, but may result in a poor approximation if some of them are missing or if we have few or low representative data for any of the available distances. Thus, since the attacker is assumed to effectuate the attack with possibly few data, in the following experiment, $\deg(P(x)) \in \{1, 3, 5\}$ have been used.

Lastly, indicate that $D_{att}^{(d)}$, which corresponds to the number of available feature observations per distance by the attacker, has been restricted to a 25% of total observations per distance.

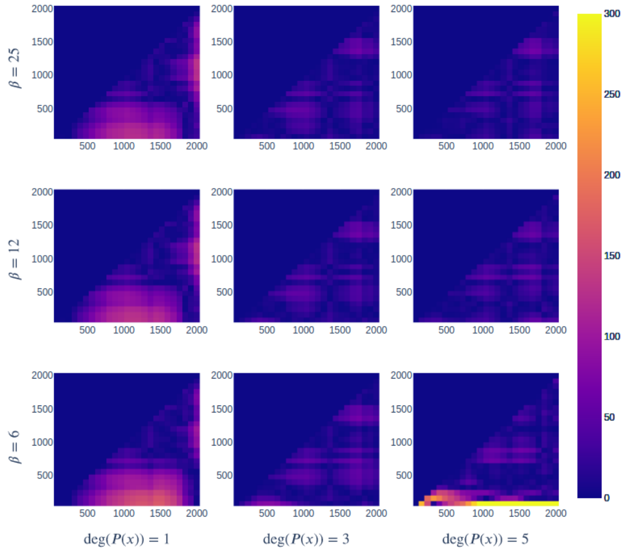

Figure 2. $\kappa_{d \to d^*}$ for the different β and $\deg(P(x))$ combinations using $D_{att}^{(d)} = 0.25 \cdot D_{test}$

The results in Figure 2 show that this proposed attack performs reasonably well, allowing us to mimic any desired distance, achieving a $\kappa^{d \to d^*}$ upper bound of 65.23 and an average value of 14.939 in the best β and $\deg((P(x))$ combination.

Regarding a more in-depth analysis of the parameters, the polynomial degree is the factor that has a greater impact on the performance of the attack. When using a linear approximation, independently of β, the transformations with $d^* - d \in [420, 1680]$ have a bigger $\kappa^{d \to d^*}$(57.683 on average) than when using $\deg(P(x)) \in \{3, 5\}$ (16.518 and 14.439 respectively).

Concerning the effects of β in the attack, it seems that the notable decrease of the parameter does not imply a proportional loss in performance for any of the tested attacks. To ratify this affirmation, we present a new comparison metric:

$$\upsilon_{p,\beta} = \frac{\sum_{d \to d^*} \kappa_{p,\beta}^{(d \to d^*)} \cdot \beta}{|d \to d^*| \cdot |d|} \tag{12}$$

where $\kappa_{p,\beta}^{(d \to d^*)}$ represents the $\kappa^{(d \to d^*)}$ obtained with β available distances and using $\deg(P(x)) = p$.

This metric aims to evaluate the cost of the attack execution taking into account the mimicking gain depending on the number of available distances. This value is highly relevant in a real scenario, where acquiring data from a certain distance has a monetary cost and generally, a fixed budget is available to perform the attack. Notice that the mean of the $\kappa^{d \to d^*)}$s has been used as the error distribution depending on β is highly similar, as shown in Figure 2.

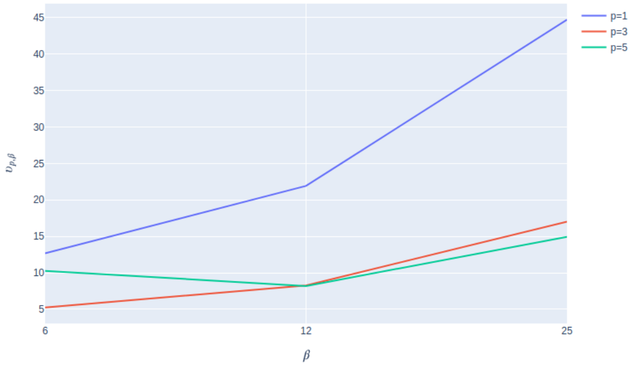

Figure 3. $\upsilon_{p,\beta}$ as a function of p and β

In Figure 3 we show how as the value of β increases, so does the cost of carrying out the attack, being especially noticeable in the last case. Thus, it can be stated that $\kappa_{p,\beta}^{(d \to d^*)}$ is not inversely proportional to β .For the tested configurations, $\beta = 6$ provides the best cost-performance trade-off except in the $p = 5$ case, which is further analyzed below.

As it can be seen Figure 2 a large error is produced when mimicking from $d = 80$ for $p = 5$ and $\beta = 6$. This is due to the fact that we are obtaining a polynomial that exactly passes through all the available feature points as a consequence of the Interpolation Theorem. This polynomial even though fitting properly for large distance, does not approximate correctly the non-seen lower ones. This phenomenon explains why in Figure 3, $\upsilon_{5,\beta}$ presents a shape different from the rest.

4.3. Symbol-to-symbol modification performance

Focusing now on the Limited Knowledge threat model, the performance of both proposed attack methods will be compared to determine whether our Gradient Decent proposal performs similarly to the Z-score one. To do so, all the possible modification ranges have been tested. Moreover, for this experiment, several $D_{att}^{(d)}$ values have been used to

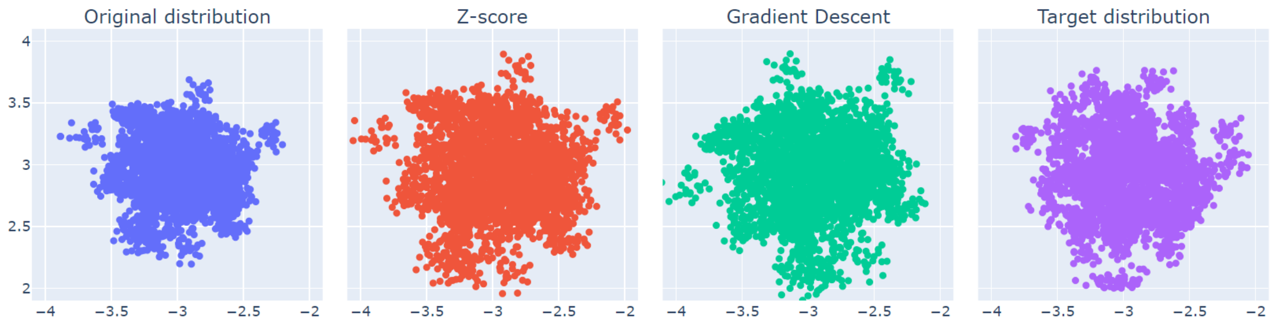

Figure 4. Example of constellation point modification ($i = 7$, $d = 1120$ Km, $d^* = 1600$ Km)

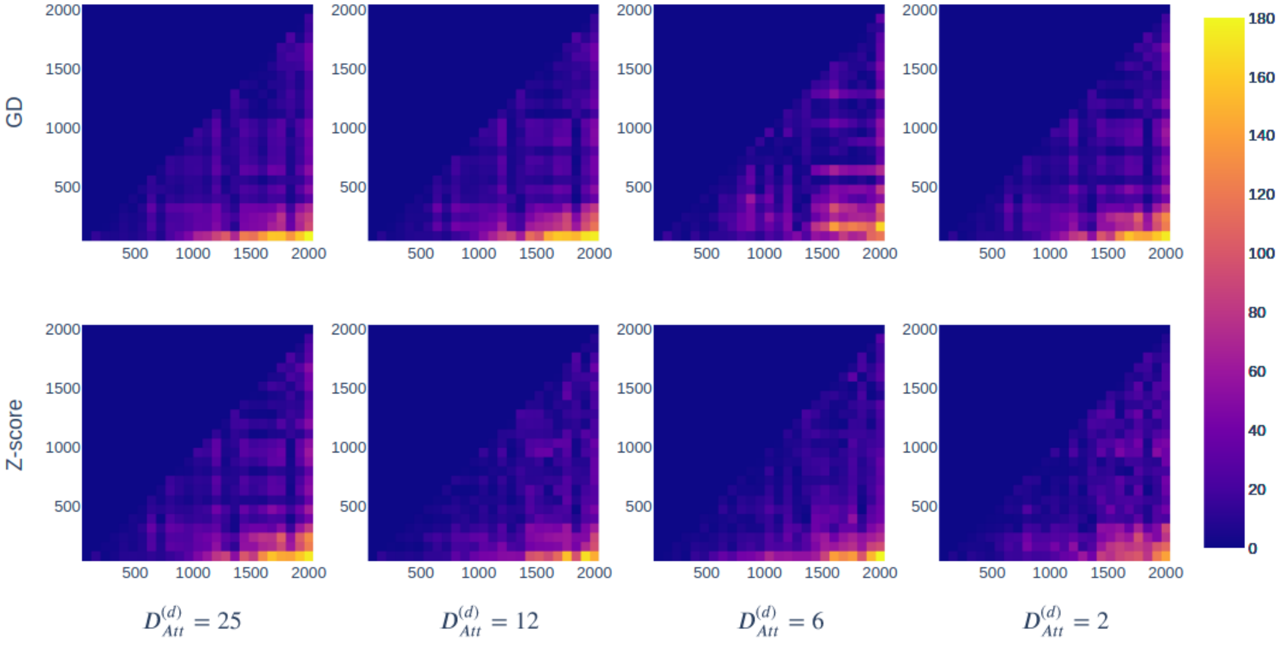

Figure 5. $\kappa^{(d \rightarrow d^*)}$ for GD and Z-score for multiple $D_{Att}^{(d)}$.

obtain insights about how the number of available observations per distance affects the quality of the NN predictions.

An example of how each method modifies a constellation point can be seen in Figure 4. Both techniques achieve a modified constellation point similar to the target one in terms of covariances and means, obtaining a practically identical symbol cloud except for a rotation. Performing a more in-depth study regarding this similarity, the difference between the features extracted from the constellations modified with GD and those obtained with Z-score is less than 10^{-5} on average over the training data. However, it is important to analyze how these small differences affect the GMM processing and the NN distance prediction.

In Figure 5 the $\kappa^{(d \rightarrow d^*)}$ achieved with both methods is presented. As seen, the distribution of the performance metric for the different $D_{att}^{(d)}$ tested is highly similar for the pair, achieving a slightly worse performance in the GD case (+14.833% error than Z-score on average).

Regarding the importance of $D_{att}^{(d)}$, the degradation in performance as a consequence of a lower observation availability for the $M_{(i)}^{(d \rightarrow d^*)}$ and $b_{(i)}^{(d \rightarrow d^*)}$ estimations is not inversely proportional, as for the β parameter in the Feature Mimic attack. The percentage of added error for lower signal sample accessibility is presented in Table 1. Indicate that the average $\kappa^{(d \rightarrow d^*)}$ for $D_{att}^{(d)} = 25$ for each method has been selected as reference, having a value of 22.703 for GD and 19.428 for the Z-score approach.

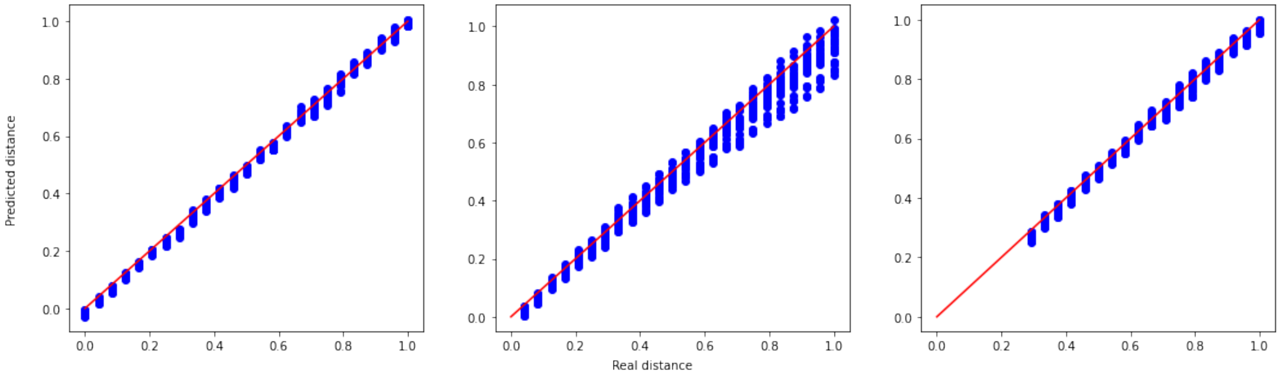

Figure 6. Normalized NN predicted distances vs.normalized real ones for clean samples(left), Z-score modified samples with $d = 80$ (middle), and Z-score modified samples with $d = 560$ (right)

Method \ $D_{att}^{(d)}$	12	6	2
GD	+5.99%	+12.17%	+20.76%
Z-Score	+1.23%	+11.35%	+29.42%

Table 1. Increase in error depending on $D_{att}^{(d)}$. $D_{att}^{(d)} = 25$ selected as reference.

Taking into account the results presented above, it can be stated that the GD approach obtains an attack performance at the same level as the Z-score method. However, both methods have a perform worse than the Feature mimic attack presented on the AK scenario.

The principal difference between the results obtained in the LK scenario and those presented in the AK one it is the range of distance that the method is able to mimic satisfactorily (*e.g.* with a $\kappa^{(d \to d^*)} < 50$). This can be especially seen when transforming with $d = 80$ to $d^* > 1000$, where the Z-score has 10 times more error than Feature mimic does.

Thus it is clear that whereas the Feature Mimic approach provides satisfactory values for any desired transformation, both symbol-to-symbol modification methods have more restrictive attack ranges, as shown in Figure 6. Therefore, it is extremely important for the attacker to be aware of this restriction before making a threat because if an excessive range of modifications is selected, the attack will not be able to overcome the security barrier.

5. Conclusions

In this work, we propose three attack strategies that try to fool a neural network-based defense system that defenses tampering attacks by predicting the distance from which a message is sent through an optical fiber. This kind of security system is fundamental to guarantee that received messages through the network are sent from the trusted source instead of somewhere in the middle of the communication channel. However, we have been able to demonstrate the effectiveness of our attacks showing a clear security vulnerability in the target defense system. Thus, further research is necessary in order to cover these vulnerabilities and ensure an effective defense method against this type of tampering attacks.

References

[1] Aleksander Madry Alexander Turner, Dimitris Tsipras. Label-consistent backdoor attacks. *arXiv:1912.02771v2*, 03 2019.

[2] Hamed Pirsiavash Aniruddha Saha, Akshayvarun Subramanya. Hidden trigger backdoor attacks. *Proceedings of the AAAI Conference on Artificial Intelligence, 34(07)*, pages 11957–11965, 2020.

[3] Fabio Roli Battista Biggio, Giorgio Fumera. Security evaluation of pattern classifiers under attack. *IEEE Transactions on Knowledge and Data Engineering, 26(4)*, pages 984–996, 2014.

[4] Agustin Louis Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. pages 536–538, 1847.

[5] Ilya Sutskever Joan Bruna Dumitru Erhan Ian Goodfellow Rob Fergus Christian Szegedy, Wojciech Zaremba. Intriguing properties of neural networks. *arXiv:1312.6199v4*, 2014.

[6] N. Costa A. Napoli J. Pedro D. Sequeira, M. Ruiz and L. Velasco. Lightweight optical constellation simulation by concatenating artificial neural networks. *2021 European Conference on Optical Communication (ECOC)*, 2021.

[7] Marija Furdek, Carlos Natalino, Fabian Lipp, David Hock, Andrea Di Giglio, and Marco Schiano. Machine learning for optical network security monitoring: A practical perspective. *Journal of Lightwave Technology*, 38(11):2860–2871, 2020.

[8] Marija Furdek, Nina Skorin-Kapov, Szilard Zsigmond, and Lena Wosinska. Vulnerabilities and security issues in optical networks. pages 1–4, 07 2014.

[9] Christian Szegedy Ian J. Goodfellow, Jonathon Shlens. Explaining and harnessing adversarial examples. *arXiv:1412.6572v3*, 2015.

[10] Sakurai Kouichi Jiawei Su, Danilo Vasconcellos Vargas. One pixel attack for fooling deep neural networks. *arXiv:1710.08864v7*, 2019.

[11] Michael Majurski Dan Kilper Uiara Celine Darko Zibar Massimo Tornatore Joao Pedro Jesse Simsarian Jim Westdorp Josh Gordon, Abdella Battou. In *Summary: Workshop on Machine Learning for Optical Communication Systems*, volume 2100, 3 2020.

[12] S. Gray K. Shaneman. One pixel attack for fooling deep neural networks. *IEEE MILCOM 2004. Military Communications Conference, 2004.*, 2004.

[13] Ambra Demontis Andrea Paudice Vasin Wongras-samee Emil C Lupu Luis Muñoz-González, Battista Biggio and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. *In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38, 2017.

[14] Matteo Russo Javier Carnerero-Cano Emil C. Lupu Luis Muñoz-González, Bjarne Pfitzner. Poisoning attacks with generative adversarial nets. *ICLR 2020 Conference*, 2019.

[15] Anthony D. Joseph J.D. Tygar Marco Barreno, Blaine Nelson. The security of machine learning. *Mach Learn 81*, pages 121–148, 2010.

[16] M. Medard, D. Marquis, R.A. Barry, and S.G. Finn. Security issues in all-optical networks. *IEEE Network*, 11(3):42–48, 1997.

[17] Muriel Médard, Doug Marquis, and Stephen Chinn. Attack detection methods for all-optical networks. 01 1998.

[18] Carlos Natalino, Marco Schiano, Andrea Di Giglio, Lena Wosinska, and Marija Furdek. Experimental study of machine-learning-based detection and identification of physical-layer attacks in optical networks. *Journal of Lightwave Technology*, 37(16):4173–4182, 2019.

[19] David Wagner Nicholas Carlini. Adversarial examples are not easily detected: Bypassing ten detection methods. *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 13–14, 2017.

[20] David Wagner Nicholas Carlini. Towards evaluating the robustness of neural networks. *2017 38th IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[21] Daniel Kowatsch Nicolas Muller and Konstantin Bottinger. Data poisoning attacks on regression learning and corresponding defenses. *arXiv:2009.07008v1*, 2020.

[22] Mausam Sumit Sanghai Deepak Verma Nilesh Dalvi, Pedro Domingos. Adversarial classification. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.*, pages 99–108, 2004.

[23] Jacob Steinhardt Pang Wei Koh and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *arXiv:1811.00741v2*, 2021.

[24] Marc Ruiz, Javier Morales, Diogo Sequeira, and Luis Velasco. An autoencoder-based solution for iq constellation analysis. In *2021 European Conference on Optical Communication (ECOC)*, pages 1–4, 2021.

[25] Siddharth Garg Tianyu Gu, Brendan Dolan-Gavitt. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv:1708.06733v2*, 08 2017.

[26] Bo Li Kimberly Lu Dawn Song Xinyun Chen, Chang Liu. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv:1712.05526v1*, 2017.