Novelty Detection: Phishing website detection

Rui Machado[1], Hélder Russa[1]

[1]Faculty of Economics, University of Porto, Porto, Portugal

Abstract. Among a lot of ways of identifying abnormal observations in a dataset, there is one method on which we will focus our study, a very specific neural network called auto encoder, also called diablo network. This technique considers a new approach, allowing to train models that simple try to predict a given input in a similar output. Even so, there are some limitations to this method when applied to novelty detection and in this article we try to exemplify how to overcome these limitations. At the end, this method will be applied to a phishing website detection context with the goal of given a trained auto encoder with normal website information, identify a phishing website as the novel case.

Keywords. Novelty detection, Auto encoder neural network, diablo network, phishing detection

1 Introduction

When you think about the context of the internet and web usage, every day millions of e-mails are sent around the globe and millions of webpages are accessed to gather, share and make transactions based on or supported by information. Due to the volume of emails and websites we consume every day, we tend assume them to be trusted and reputable, however not all of them are legit and a growing number of cyber-thieves are using the same systems to steal private information with the objective of monetize it. This is often called a phishing attack. [1]

Marsland [2] classified Novelty detection as a method for detection of unforeseen or abnormal events that a machine learning system has not been trained with and was not previously aware of. When you think about the phishing attack context, there is a high volume of normal data, related with the legit email and website accesses that are made every day, while the phishing attack data is still small and scarce. With this in mind, one can assume Novelty detection as a good method for detection of such events and help fighting the phishing based cyber crimes.

This article is organized in three chapters. The first part will be to explain what is Novelty Detection, some guideline and principles to build such a model and the main families of techniques. In the second part we will apply a chosen novelty detection technique, namely an Auto encoder, to a phishing detection context. The aim of this study is to train a model that is able to accurately identify a phishing website.

2 Novelty Detection

The concept of unforeseen or abnormal, novel, event is a relative concept regarding the problem domain knowledge and must be defined in the context of the representation of our current problem. It is a useful technique in cases where an important class of data is under-represented in the training set. In some cases, like in medicine, a novel case might be a disease or potential health issue that will make doctors act on the patient and when no novel cases are detected, no reaction is needed. [2]

When taking the defining of novelty detection in consideration, one might find it a similar concept to the so called outlier detection. In fact, these two concepts are closely related, even if the purpose of usage differ, we might differentiate them by taking the dimensional level in consideration. While outlier detection methods can be used to identify observation points that are considerably distant from other observations in a univariate perspective, it is not possible, using a generally applicable, parameter-free method, to find an outlier in multivariate (high-dimensional space) data by examining the variables one at a time. [2]

Novelty detection is one of the fundamental requirements of a good classification system. Because it's impossible to know, beforehand, all the possible object classes and train a classification model with all of them, the performance of the predictions for the under-represented classes will be poor. Given this a good classification system must have the ability to differentiate between known and unknown objects during testing phase. For this purpose, a lot of different models for novelty detection have been proposed by researchers, based in either statistical or machine learning approaches. The main reason for the amount of options and methods for novelty detection is because the success of its application depends largely on the statistic properties of the data being handled, so different authors have been working on different types of data, so currently a wide variety of methods cover a wide set of types of data and problems. [3]

2.1 Principles for a good Novelty Detection System

Because of the complexity surrounding novelty detection systems Markou & Singh, (2003) have defined a set of seven principles that should be respected to maximize the accuracy and performance of such systems, namely:

1. *Principle of robustness and trade-off*: A novelty detection method must be capable of robust performance on test data that maximizes the exclusion of novel samples while minimizing the exclusion of known samples. This trade-off should be, to a limited extent, predictable and under experimental control.
2. *Principle of uniform data scaling*: Standardization should be applied to data so that both test data and training data lie within the same range;
3. *Principle of parameter minimization*: Dimension reduction should be applied so that the minimum number of interest variables are used;
4. *Principle of generalization*: The system should be able to generalize without confusing generalized information as novel;

5. *Principle of independence*: The system should be independent of the number of features, and classes available and it should show reasonable performance in the context of imbalanced data set, low number of samples, and noise;
6. *Principle of adaptability*: The system when recognizing novel samples during test should be able to use this information for retraining;
   Principle of computational complexity: A number of novelty detection applications are online and, therefore, the computational complexity of a novelty detection mechanism should be as low as possible.

According to Pimentel, Clifton, Clifton, & Tarassenko (2014) Novelty detection methods (also called approaches) can be organized in five different types namely probabilistic types, distance-based, reconstruction-based, domain- based, and information-theoretic techniques. The next subchapter intends to explore on a summarized perspective each of these types, helping understanding their structure and when to use each of them.

2.2 Probabilistic novelty detection

Probabilistic or also called statistical methods define a family of novelty detection methods that are based in the assumption that normal data are generated from an underlying data distribution, which may be estimated from example data and abnormal observations don´t fit the same distribution. These methods start by estimating a probability density function of the data, which may be thresholded to delimit normal and abnormal areas of data space, and based on it test if a given test sample was generated from the distribution or not. [4]

Statistical methods can be both parametric and non-parametric. When parametric they assume that data was generated from an underlying distributions and the parameters of the distribution estimated from observed data (Such as mean and standard deviation). In order to reduce the risk of error with this type of novelty detection methods, techniques like Gaussian Mixture Models[1] can be used. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm[2] or Maximum A Posteriori (MAP)[3] estimation from a well-trained prior model. Another application for this type of methods is for novelty detection of time-

[1] A Gaussian mixture model is a probabilistic model that assumes that all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. These models try different number of distribution combinations until a good fit is found. [5]

[2] Expectation-Maximization is a density estimation algorithm that enables parameter estimation in probabilistic models with incomplete data. (Unlabeled data can be classified as a nice case to use this algorithm). [12]

[3] Maximum A Posteriori is a density estimation algorithm that allows to incorporate prior information about the parameters distribution in the training process. [13]

series data, and one of the best well known methods is the Hidden Markov Models (HMMs)[4] which provide a state-based model of a data set. [5]

Non-parametric methods do not assume any distribution on a give data sample and again multiple techniques can be found to fit the needs and complexity of the problem being analyzed. One of the techniques that can be found in this family of novelty detection methods is Kernel Density estimators, in which the probability density function is estimated using a large number of kernels distributed over the data space. This technique, also called *Parzen windows* estimator is important for the topic in study, Intrusion detection, as its one of the most used technique for this purpose. [4]

2.3 Distance-based novelty detection

These methods start from the same structure as the probabilistic methods when it comes to generate a probability density function of the data, however they use defined distance metrics instead of statistical techniques to compute the similarity between two data points and consequently based on a given distance threshold, state if we are observing a novel case or not. There are two main approaches for distance-based novelty detection, namely, clustering and nearest neighbor based. [4]

Starting with the nearest neighbor-based approaches, the rationale behind them is that normal data points have a considerable number of close neighbors while novel points are located far from those points. For the calculation of the distance measure several methods can be used like Euclidean distance, Mahalanobis distance or any other, based on the type of variables we are handling. [4]

Another technique that can be found in this family is clustering based novelty detection. In clustering techniques data is partitioned in to a number of clusters, typically using the distance between data points. With this in mind, a novel case can be detected when a given data point doesn't belong to any of the available clustering classes.

Because these methods based in calculating distances between points have some lack of performance when handling high-dimensional data, some statistical techniques like Principal Component Analysis can be used for dimensionality reduction allowing to discard variables that explain less than a given threshold of variance. [3]

2.4 Reconstruction-based novelty detection

Reconstruction-based novelty detection, is normally associated and implanted using Neural Networks for which auto encoder networks have been widely used for detecting novel cases. The idea behind this technique is that the network is trained using only normal behavior observations and when a given observation falls outside the range of data the network has been trained with, its classified as a novel case. The structure of an auto encoder consists on the same number of inputs and outputs and the training

[4] Hidden Markov Models are statistical Markov models in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. An HMM can be presented as the simplest dynamic Bayesian network.

phase attempts to match each output to the corresponding input, so that the error between inputs and output is low for the normal data but varies when the input dataset is unrecognized. There are several benefits when using this technique, such as data compression and incorporated feature selection with PCA which help improving the performance of executing novelty detection. [6]

This method in parallel with the probabilistic *Parzen windows* estimator is also commonly used by the community for intrusion detection. Marais & Marwala (2003) are one example, of study and implementation of auto-encoders for predicting the presence of Internet Worms using Novelty Detection.

2.5 Domain-based novelty detection

In this type of novelty detection techniques, the purpose changes, so instead of estimating a density function or leaning a neural network, we intend to separate the target class boundary (domain boundary) in such a way that all observations outside this boundary are novel cases. One of the most popular techniques in this family is the so called "One-class SVM", which is an unsupervised algorithm that learns a decision function able to classify new data as similar or different to the training set. [3]

By just providing the normal training data, this algorithm is able to create a representational model of it. When new observations are evaluated in the trained model, if its results prove to be too different, it is labeled as out-of-class. [7]

2.6 Information-theoretic novelty detection

Information-theoretic novelty detection techniques are based on entropy[5] and states that novel cases change significantly the value of the information content when compared to normal data. [3]

As in the previous families described, the main idea behind this family of techniques is to measure the impact of a test sample on the previously learnt model, and is the entropy varies significantly we can infer that we are observing a novel case. [8]

[5] Entropy in information theory is a measure of the uncertainty associated with a random variable.

3 Using auto encoders for phishing website detection

3.1 Phishing

The major goal of one phishing attack is steal user identities and credentials. This kind of attack is based on pretending to be someone else that is not, in order to obtain sensitive information from the victims seducing them into visiting fraudulent websites and persuading to enter identity information such as usernames and passwords, personal identification numbers (e.g. credit card number) and any additional information that can be made to seem relevant for the attacker. [9]

According to [10], on their Phishing Activity Trends Report from 4th Quarter 2015, the unique type of phishing email attacks has increased into the hundreds of thousands of discrete types of attacks per year. Only during the fourth quarter of 2015 the phishing attacks for the Retail/Service and Financial sector was with the highest rate with almost 50% of all phishing attacks.

Currently one of the most common ways of phishing attack is made by spamming email messages that can be a future account suspension, a payment for a marketing survey, or a bank transaction that the user knows to be fake and needs to cancel it. When a victim receives that email it is misled to believe that is legit and needs to make some additional actions inside the Webpage that usually is visually an exact replica of an official Webpage.

It is important to distinguish phishing messages from legitimate messages, since as told, phishing messages are built to appear genuine as much it can, so to identify phishing messages one can use defense techniques such as the use of collaborative filtering (e.g. user reports), blacklisting (e.g. URLs collected from honeypot email accounts) or text classification combined with similarity testing methods (e.g. minor variations of known phishing messages). Even though the mentioned techniques can be effective after an initial positive identification, they require a few time where some percentage of the phishing messages reach the victims and are positively identified. Due to the similarity between phishing and legitimate messages, spam-filter style text classification filters have difficulties working alone and are typically used in combination with similarity tests and blacklists, however as it is expected, phishing message creators are always changing their message format and content when they realize that the post message were discovered/identified or blacklisted making this a constant evolution of phishing attacks resulting in a race between the creation of new phishing messages and defenses used to identify them. [11]

3.2 Auto encoder for phishing website detection

Using auto encoders to detect phishing websites, which will be the novel cases, requires firstly we train our neuronal network using only normal behavior – expected attributes that characterize some legit website. With our trained model, when data dissimilar to the training data is presented, the network must be able to identify the novelty.

Essentially the basic structure of an auto encoder to detect a phishing website remains the same as when we try to use it for other scopes, we need to have the same number of inputs as outputs. Training requires teaching the network to attempt to match each output to the corresponding input, so that the error between inputs and outputs is low for the normal or training data, but varies when the input dataset is unrecognized. [6]

3.2.1 Training and testing data

The used dataset was collected from the data repository of the University of California *(UCI Learning Repository Machine)*, and can be found under the name " Phishing dataset sites". This set of around 10.000 observations contains a collection of variables the authors identify as key to the detecting a phishing website, such as website address length, external sources inside HTML code, of amongst others. Mohammad, Thabtah, & Mccluskey (2013)

3.2.2 Development

To develop this novelty detection model, we have used the R platform, which though the library "*auto encoder",* developed by in Eugene Dubosarsky and Yuriy Tyshetskiy, allowed us to train an auto encoder and predict outputs for new observations. The following pseudo-code block reflects the implementation needed to train an auto encoder using this library.

```
Inputs: Training data set trn, Number of layers nl,
Network unity type ut, Number of neurons in the hidden
layer nh, A weight decay parameter lambda, A weight of
sparsity penalty term beta, Weights initialization
epsilon, Sparsity parameter rho, Number of max iterations
maxit, An optimization method opm
    Initialize: trn ={training set}, nl=3, ut="logistic",
    nh =number_of_columns(trn, lambda=0.0002, beta=6,
    epsilon=0.001, rho=0.01, maxit =2000, opm = 'BFGS'
      call autoencode(trn, nl, nh, ut, lambda, beta, rho,
      epsilon, maxit, opm)
```

```
Output: A trained auto encode neural network object
containing a list of weight matrices W, a list of biases
b, A unit type unit.type, a rescaling list of element
rescaling, An average mean error for the training set
mean.error.training.set.
```

After training the auto encoder we can now use it to predict new observations, which in practice results in a dissimilarity matrix between the given input variables and the output it has predicted.

For the testing and evaluation phase of our trained model, we have decided to follow a cross validation based approach were we test different folds but of different sizes as

well to have a scalability performance evolution. The following flux gram summarizes the tasks made in the testing/prediction phase to be able to identify novel cases.

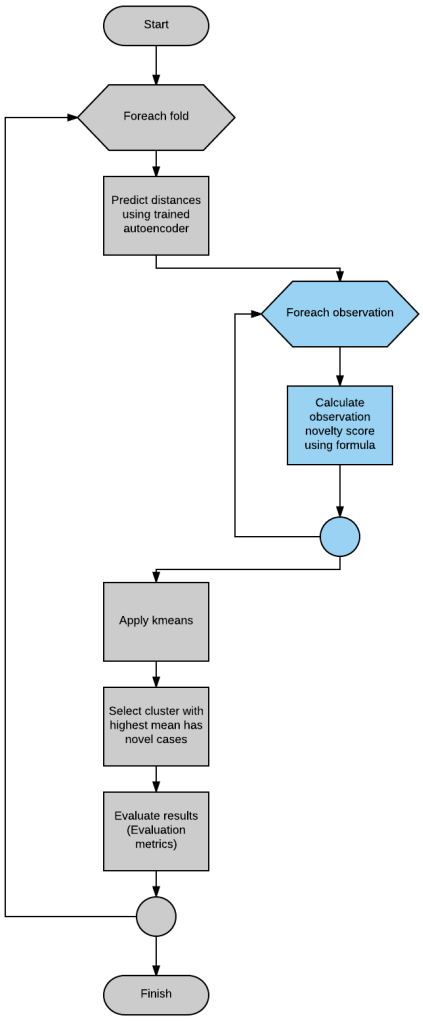

Figure 1 - Flux gram for the used model of identifying novel cases

When predicting an output for a given observation, the auto encoder returns a list of values, correspondent to the deviation of each input to the corresponding output. With this in mind we had to come up with a solution to attribute a novelty score to the observation as a whole using the several deviations of its inputs. To do so, we used the same approach suggested by Marais & Marwala (2005), where we calculate it using the sum of the mean squares of the outputs. The formula used is described below and the basic idea is that for each observation we sum the differences between each of its output and the correspondent input, squared, and at the end you divide by the number of observations, getting at the end the mean squared distance per observation.

$$\frac{1}{n} \times \sum_{1}^{n} (x'_n - x_n)^2$$

After calculating a novelty score for each predicted observation, we can now analyze the average mean squared distances of these classifications and define a threshold which can work has a limit to separate normal from novel cases (The higher the distance between output and input the higher the probability of being a novel case). As you can see in Figure 1, it might not be simple to define such a threshold manually and visually.

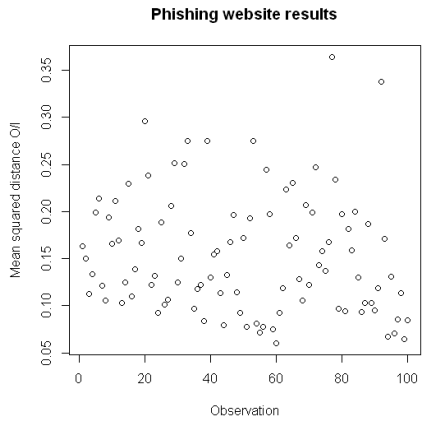

Figure 2 - Result of plotting the mean squared distances per test observation

Because of this difficulty, we would like to convert this task into an automated one. This way we have decided to combine our auto encoder predictions distance results with a K-means classification, forcing it to have just two centers, which in theory would split the observation into normal and novel cases. The way we have implemented it is that after running a k-means clustering technique with two centers we calculate the mean value for both clusters and take the one with the highest as the novel cases clusters. The reason we have decided this is because if the mean is higher in such cluster that means that the deviations between outputs and inputs are higher in this cluster which by definition can we used to classify novel cases.

3.2.3 Results analysis

After training the auto encoder neural network, we applied on a cross validation approach, a set of predictions, for which the results can be observed in the following Table 1.

Test cases	% Train	TPR	TNR	ACC	ERR	False Alarm	Missed
25	1%	89%	75%	80%	20%	11%	25%
50	1%	95%	71%	80%	20%	5%	29%
100	2%	95%	77%	84%	16%	5%	23%
150	3%	94%	72%	79%	21%	6%	28%
200	4%	95%	65%	74%	26%	5%	35%
250	5%	96%	63%	72%	28%	4%	37%
300	6%	96%	60%	70%	30%	4%	40%
500	10%	94%	62%	73%	27%	6%	38%
1000	20%	93%	62%	73%	27%	7%	38%
5000	102%	91%	63%	73%	27%	9%	37%
11055	226%	90%	63%	72%	28%	11%	37%

Table 1 - Novelty detection accuracy using the trained auto encoder

The main observations show that the generated auto encoder has a high rate of true positive predictions, varying between 89% and 96%, meaning its quite good predicting novel cases. However, in term of true negative rate, comes with a lower performance, varying between 62% and 77%.

In other words, we can use these two variables to interpret our accuracy in detecting a phishing website or not, by creating a false alarm rate and missing phishing website rate. If we take the TPR as a first analysis variable, the remaining percentage until we explain 100% can be defined as false alarm rate, as it's the percentage of observations we are classifying as novel cases but are not. If we look into the table, we can calculate an average of 7%, which we consider a very good result.

On the other hand, if we apply the same logic on the TNR, we obtain all the novel cases that should be classified as such, but were considered normal, meaning it's a missed phishing website detection. If we look into our results, we obtain as average of 33% which is a relatively high value.

The following Table 2, summarizes the average values observed in the evaluation of our novelty detection model.

Avg. TPR	Avg. TNR	Avg. Accuracy	Avg. Error Rate	Avg. False Alarm Rate	Avg. Missed Phishing Rate
93%	67%	75%	25%	7%	33%

Table 2 - Summary of model evaluation analysis

4 Conclusions

During this work we applied novelty detection, based on an auto encoder technique to train a model able to identify a phishing website, with a practical implementation of algorithms in R. Although the results were good in terms of true positive rate, we believe that to be used in a real scenario, we had to improve the true negative rate and thus avoid having such a big number of missing cases. This way we cannot classify the result as an extraordinary result but rather a positive indication that this technique might bring good results if further improvements can be made.

We can also verify that when training an auto encoder, the big bottleneck is at prediction time, namely when it comes to dynamically and automatically define a good threshold to decide whether a prediction is a novel case or not. Regarding this we believe it's a good future work suggestion of researching and testing the implementing of different techniques for such purpose that might bring better results.

In fact, the model training itself involves the tuning of multiple parameters that end up affecting the quality of the trained model. These parameters can be further explored and tested different variations combined with a better prediction threshold definition can improve the overall accuracy of our phishing attack detection system.

As a future work suggestion we would like to leave a note for the multi agent systems domain, where if a set of agents are equipped with a previously trained auto encoder for different phishing scenarios data, a company can implement a proactive phishing email or website detection system.

5 Bibliography

[1] A. Elledge, “Phishing: An Analysis of a Growing Problem,” 2007.

[2] S. Marsland, “Novelty Detection in Learning Systems,” *Neural Comput. Surv.*, vol. 3, pp. 1–39, 2002.

[3] M. Markou and S. Singh, “Novelty detection: A review - Part 1: Statistical approaches,” *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[4] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, “A review of novelty detection,” *Signal Processing*, vol. 99, pp. 215–249, 2014.

[5] D. A. Clifton, L. A. Clifton, P. R. Bannister, and L. Tarassenko, “Automated Novelty Detection in Industrial Systems David,” *Methods*, vol. 296, pp. 269–296, 2008.

[6] E. Marais and T. Marwala, “Predicting the Presence of Internet Worms using Novelty Detection,” 2003.

[7] R. Vlasveld, “Introduction to One-class Support Vector Machines,” *Roemers Blog*, 2013. [Online]. Available: http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/.

[8] P. Bodesheim, F. Schiller, E. Rodner, A. Freytag, and J. Denzler, “Divergence-Based One-Class Classification Using Gaussian Processes,” *Procedings Br. Mach. Vis. Conf. 2012*, pp. 50.1–50.11, 2012.

[9] C. B. Do and S. Batzoglou, “What is the expectation maximization algorithm?,” *Nat. Biotechnol.*, vol. 26, no. 8, pp. 897–899, 2008.

[10] J. L. Gauvain and C. H. Lee, “Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains,” *Speech audio Process. ieee ...*, no. April, pp. 1–16, 1994.