# Statistical Evaluation of Artificial Intelligence -based Intrusion Detection System

Samir Puuska, Tero Kokkonen, Petri Mutka, Janne Alatalo, Eppu Heilimo, and Antti Mäkelä

Institute of Information Technology, JAMK University of Applied Sciences, Jyväskylä, Finland
{samir.puuska, tero.kokkonen, petri.mutka, janne.alatalo, eppu.heilimo, antti.makela}@jamk.fi

**Abstract.** Training neural networks with captured real-world network data may fail to ascertain whether or not the network architecture is capable of learning the types of correlations expected to be present in real data.
In this paper we outline a statistical model aimed at assessing the learning capability of neural network-based intrusion detection system. We explore the possibility of using data from statistical simulations to ascertain that the network is capable of learning so called precursor patterns. These patterns seek to assess if the network can learn likely statistical properties, and detect when a given input does not have those properties and is anomalous.
We train a neural network using synthetic data and create several test datasets where the key statistical properties are altered. Based on our findings, the network is capable of detecting the anomalous data with high probability.

**Keywords:** Statistical Analysis, Intrusion Detection, Anomaly Detection, Network Traffic Modeling, Autoregressive Neural Networks

## 1 Introduction

Neural networks are being increasingly used as a part of Intrusion Detection Systems, in various configurations. These networks are often trained in ways that include both legitimate and malicious recorded network traffic. Traditionally, a training set is used to train the network, while another set of samples is used to assess the suitability of the proposed architecture. However, further assessment of the network architecture depends on knowing what statistical properties the network can learn, and how it will react if these properties change.

In this paper, we present a way to estimate if a network has the capability of learning certain desired features. Our analysis approach is to ascertain that the network can learn precursor patterns, i.e. patterns that are necessary but not sufficient conditions for learning more complex patterns of the same type. The goal is to supplement traditional sample-based learning with synthetic data

variants that have predictable and desirable statistical properties. This synthetic data can then be used both to increase the dataset and to address known biases that often arise when collecting real-world data traffic.

Certain real-life phenomena, such as network traffic, can be considered to have known intrinsic properties due to their artificial nature. In communication protocols, for example, certain hard limits must be observed for achieving any successful communication. Although protocols are sometimes abused for malicious purposes, there are still limits as to how extensive the effect can realistically be. On other occasions, there are limits on how much any given feature can be expected to correlate with anomalies. However, a combination of these weakly-correlated features may, if they form a specific pattern, signal for an anomaly. In artificial systems, it is sometimes possible to distinguish correlation from causation, and therefore make more intelligent predictions by considering only the direction that is actually feasible.

Based on their basis of analysis, there are two classes of Intrusion Detection Systems (IDS): anomaly-based detection (anomaly detection) and signature-based detection (misuse detection). Anomaly-based detection functions without earlier gathered signatures and are effective even for zero-day attacks and encrypted network traffic. There are various machine learning techniques implemented for classifying anomalies from network traffic but still, some flaws exist; a high amount of false alarms and low throughput [2, 6, 5].

In our earlier studies, we implemented two anomaly-detection based IDSs that utilized deep learning. Our first model was based on wavelet transforms and Adversarial Autoencoders [8]. That model was improved with a WaveNet [7] based solution [4]. In this paper, we perform a statistical experiment for determining the performance of a WaveNet based IDS system.

## 2 Method

We begin by outlining a statistical model which complies with our research goals. As stated, the idea is to construct a statistical distribution which contains so-called precursor or proto-elements of the actual phenomenon. The aim here is to ascertain that the network is capable of learning simpler versions of the relationships expected to be present in the real data.

Network protocols have a certain degree of predictability. As previously stated, we can state certain hard limits for the features we have selected. Our model is designed to work with the Transport Layer Security (TLS) protocols, as encrypted HTTP traffic is a common communication channel for malware.

We can identify various types of noise that usually occurs in the networks. The model should be resistant to this type of noise, as we know it arises due to the nature of data networks and is likely not associated with the type of anomalies we are interested in.

Based on this reasoning we have constructed a model that incorporates three distributions modeling i) packet size, ii) packet direction, and iii) packet timings.

One packet is modeled using these three features. The packet structure is illustrated in Figure 1. A connection consists of 250 packets (vectors), where timings are expressed using time differences to the next packet.

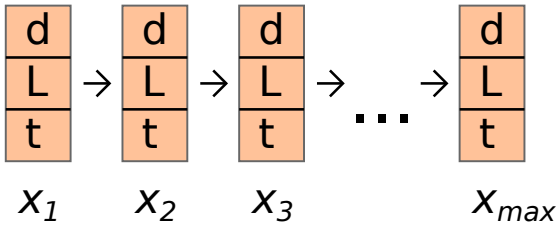


Fig. 1: Visual description of a single connection. Each packet consists of a vector that contains three elements: packet direction, packet size, and time difference to the next packet.

### 2.1 Packet size and direction

Based on the findings by Castro et al. [1], we model the packet size using the Beta distribution ($\alpha = 0.0888$, $\beta = 0.0967$). We enforce two strict cut off points: the minimum (15) and maximum (1500). This reflects the packet size constraints that networking protocols impose on packet size.

Packet direction is determined using the packet size. This models the real-world phenomenon where the requests are usually smaller than the responses. In the model packet direction, there is a binary value determined by packet size $L$; packets smaller than 30 are outgoing and larger than 1200 are incoming. If the size is $30 < L < 1200$, the direction is decided randomly.

### 2.2 Packet timing

Various separate processes affect packet timing: the nature of the protocol or data transfer type determines how fast packets are expected to be sent or received. For example, fetching an HTML page via HTTP creates a burst of packets going back and forth; however, malware that polls a Command and Control server at late intervals (for example hourly) may send just one packet and get one in response. However, a considerable amount of variance is expected when systems are under a high load or there is a network issue. Therefore, not all anomalies in the timing patterns are malicious in nature.

Since we do not need to model the traffic explicitly, we use a packet train model [3] inspired composite Gaussian distribution model for creating packet timings. Originally, the packet train model was designed for categorizing real-life network traffic, not for generating synthetic network data.

For the relevant parts, the packet train model is characterized by the following parameters; *mean inter-train arrival time*, *mean inter-car arrival time*, *mean train-size*. We capture the similar behavior by combining two normal probability density functions in range $x \in [a, b]$ as:

$$f(x) \equiv f(x; \mu_1, \mu_2, \sigma_1, \sigma_2, w_1, w_2, a, b) = \begin{cases} 0 & \text{if } x < a \\ \frac{R}{\sqrt{2\pi}} \left[ \frac{w_1}{\sigma_1^2} \exp\left( -\frac{(x-\mu_1)^2}{2\sigma_1^2} \right) + \frac{w_2}{\sigma_2^2} \exp\left( -\frac{(x-\mu_2)^2}{2\sigma_2^2} \right) \right] & \text{if } a \leq x \leq b \\ 0 & \text{if } x > b \end{cases} , \quad (1)$$

where $R$ is normalization constant that is calculated from normalization condition for total probability. In (1), $\mu_1$ and $\mu_2$ are mean values for sub-distributions ( $\mu_1 < \mu_2$) , and $\sigma_1$ and $\sigma_2$ are relevant variances. Sub-distributions have relative weights $w_1$ and $w_2$.

We chose to use a semi-analytical probabilistic model since it is easier to parameterize and understand than more generic Markov models. Our model captures the most relevant properties of the train packet model; roughly *mean inter-train arrival time* $\propto \mu_2$, *mean inter-car arrival time* $\propto \mu_1$, and *mean train size* $\propto w_1/w_2$. Corresponding cumulative distribution function can be expressed with complementary error functions and solved numerically for generating random number samples with desired statistical properties.

### 2.3 Scoring

Since our neural network is trained by minimizing the mean of minibatches discretized logistic mixture negative log-likelihoods [9], we can detect the anomalous connections by observing the mean negative log-likelihood of the feature vectors in a single sample. Moreover, we introduce the different types of anomalies in varying quantities to the dataset to evaluate the neural network's sensitivity and behavior.

### 2.4 Tests

We trained the neural network using data formed by previously described clean distributions. The size of the training set was 160000 samples. We reserved an additional 40000 unseen samples for the evaluation.

The test procedure consists of generating samples where the parameters are drawn from a different distribution than the training data. These "anomalous" samples are mixed with the evaluation data to form ten sets where the percentage is increased from 10% to 100%. Each of these datasets is evaluated using the neural network and the changes in the mean anomaly score are observed in Table 1, which describes the three types of alterations made to the samples. The alterations were chosen because they represent different correlations; namely, the directionality is determined between two features inside one packet, whereas

the change in timing distribution is spread out between packets and does not correlate with other features inside a particular vector. This approach is expected to test the network's capability to detect both kinds of correlations.

| Test | Description |
|---|---|
| Direction | This test swaps the directionality decision criteria. Small packages are now incoming and large outgoing. The area where the directionality is randomly determined stays the same. |
| Time | This test replaces the bimodal distribution on packet timing with unimodal Gaussian distribution $\mu = 50$, $\sigma = 80$. The cut-off points remain the same. |
| Combined | The test combines both alterations to the dataset. |

Table 1: Descriptions of the alterations.

## 3 Results

The results indicate that the network learned to detect anomalous data in all three datasets. The results are illustrated in Figure 2. As the figure indicates, the anomaly score keeps increasing with the percentage of "anomalous" data.

Packet direction seems to have an almost linear increase in the anomaly score, whereas changes in time distribution result in a sudden jump, after which the score keeps increasing relatively modestly. The combined data exhibits both the starting jump and the linear increase. This is a desired outcome, as it indicates that the anomaly score reflects the change in data in a stable fashion.

In summary, the network was able to learn the properties outlined in the previous sections. The results indicate that the network can detect correlation inside the vector, as well as between vectors. This outcome supports the notion that a neural network structured in this fashion learns useful relationships between the features.

## 4 Discussion

When constructing a machine learning solution for anomaly detection, the available data may not be suitably representative. This situation may arise, for example, when collecting or sampling the dataset in a statistically representative way is impossible for practical reasons. It is not feasible to expect a statistically representative sample of all possible network flows, even when dealing with one application. Moreover, the data in networks may exhibit correlations known

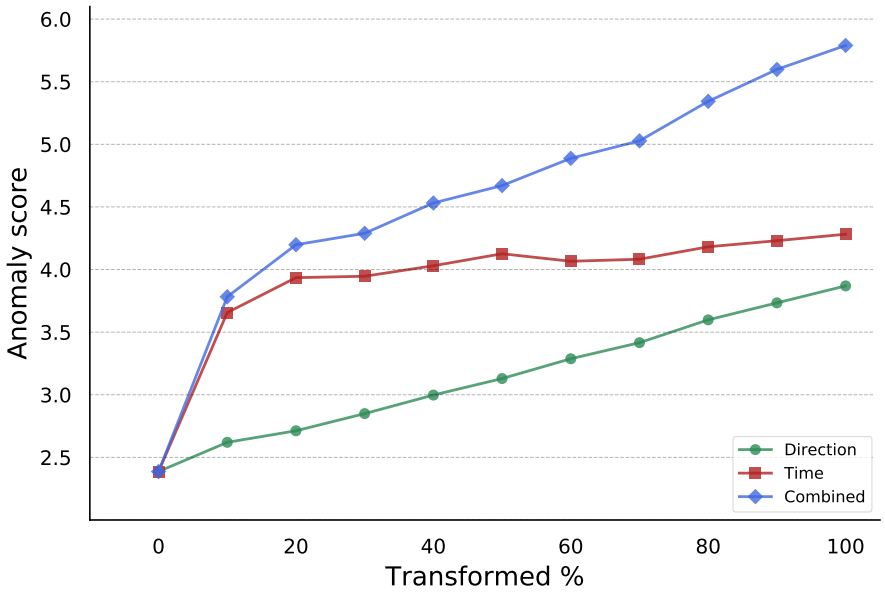


Fig. 2: Plot of test results from each anomaly type. The horizontal axis indicates the percent of samples that were altered. As expected, the mean anomaly score on the vertical axis increases with respect to the amount of altered samples in the test data.

to be unrelated to the type of the anomaly under examination. The statistical properties of network data may fluctuate due to multiple factors.

By using synthetic data which contains correlations that are known to be relevant, it is possible to verify whether or not the proposed network structure is capable of detecting them in general. Moreover, the test may show how the classifier reacts to the increase in variance. In an ideal case a classifier should be relatively tolerant to small fluctuations; however, be able to reliably identify the anomalous samples.

Future work includes refining the statistical procedures, as well as increasing the complexity of correlations in test data. Further research will be conducted on how the relationship between increasing variance and data are drawn from different distributions affects the anomaly score, and how this information may be used to refine the structure of the neural network classifier.

## Acknowledgment

This research is funded by:

– *Using Artificial Intelligence for Anomaly Based Network Intrusion Detection System* -project of the Scientific Advisory Board for Defence (MATINE)

– *Cyber Security Network of Competence Centres for Europe (CyberSec4Europe)* -project of the Horizon 2020 SU-ICT-03-2018 program

## References

1. Castro, E.R.S., Alencar, M.S., Fonseca, I.E.: Probability density functions of the packet length for computer networks with bimodal traffic. International journal of Computer Networks & Communications **5**(3), 17–31 (2013). DOI 10.5121/ijcnc.2013.5302
2. Chiba, Z., Abghour, N., Moussaid, K., Omri, A.E., Rida, M.: A Clever Approach to Develop an Efficient Deep Neural Network Based IDS for Cloud Environments Using a Self-Adaptive Genetic Algorithm. In: 2019 International Conference on Advanced Communication Technologies and Networking (CommNet), pp. 1–9 (2019). DOI 10.1109/COMMNET.2019.8742390
3. Jain, R., Routhier, S.: Packet trains–measurements and a new model for computer network traffic. IEEE Journal on Selected Areas in Communications **4**(6), 986–995 (1986). DOI 10.1109/JSAC.1986.1146410
4. Kokkonen, T., Puuska, S., Alatalo, J., Heilimo, E., Mäkelä, A.: Network anomaly detection based on wavenet. In: O. Galinina, S. Andreev, S. Balandin, Y. Koucheryavy (eds.) Internet of Things, Smart Spaces, and Next Generation Networks and Systems, pp. 424–433. Springer International Publishing, Cham (2019)
5. Masduki, B.W., Ramli, K., Saputra, F.A., Sugiarto, D.: Study on implementation of machine learning methods combination for improving attacks detection accuracy on intrusion detection system (ids). In: 2015 International Conference on Quality in Research (QiR), pp. 56–64 (2015). DOI 10.1109/QiR.2015.7374895
6. Narsingyani, D., Kale, O.: Optimizing false positive in anomaly based intrusion detection using genetic algorithm. In: 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE), pp. 72–77 (2015). DOI 10.1109/MITE.2015.7375291
7. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: Wavenet: A generative model for raw audio (2016). URL https://arxiv.org/pdf/1609.03499.pdf
8. Puuska, S., Kokkonen, T., Alatalo, J., Heilimo, E.: Anomaly-based network intrusion detection using wavelets and adversarial autoencoders. In: J.L. Lanet, C. Toma (eds.) Innovative Security Solutions for Information Technology and Communications, pp. 234–246. Springer International Publishing, Cham (2019)
9. Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017 (2017). URL https://openreview.net/references/pdf?id=rJuJ1cP_l