Hybrid deep-learning analysis for cyber anomaly detection

Stanimir Kabaivanov¹, Veneta Markovska²,*

¹ Plovdiv University “Paisii Hilendarski”, Plovdiv, Bulgaria
² University of Food Technologies, Plovdiv, Bulgaria

* venetta@abv.bg

Abstract. Cyber threats evolve continuously and so do the detection tools and algorithms. In this paper we analyse the efficiency of hybrid deep-learning analysis as a mean to detect anomalies in computer network traffic. Different deep-learning algorithms are tested against real network intrusion events in an attempt to assess their potential as an early warning system. We suggest a combination of algorithms and rule-based filters as a hybrid system that can improve efficiency and accuracy of cyber anomaly detection.

1. Introduction

Cyber threats are not new and it often seems like we have already done enough to prevent them by installing state-of-the-art protection system and “sanitizing” our online behavior. Recent reports however indicate that although attacks do not grow at a very fast rate they seem to get more sophisticated and indirect ([1]). Taking into account the global move toward even more distributed working environment in the wake of the COVID-19 crisis, the potential for destructive and sophisticated attacks has grown significantly. Some of the possibilities are well known – for example exploitation of cloud vulnerabilities and social engineering attacks. Belani in [2] points out that artificial intelligence, adaptive attacks and machine learning will be major issues in the years to come.

In a distributed environment detecting a malicious software after it has been deployed and activated is a costly and definitely late measure. It makes much more sense to try to monitor the environment and react on anomalies in the normal/expected operation, instead of passively wait for a problem to appear and then try to resolve it. Considering the high amount of data and increasing number of connected software applications such monitoring requires to have dedicated solutions and automated filtering of possible problems. Without proven and solid algorithms that point our attention only to real dangers we will be overwhelmed by false positive alarms that eventually will cause more harm than good.

Bhuyan et al in [3] classify anomaly detection methods into eight broad groups – statistical, classification, soft-computing, clustering, ensemble, fusion and hybrid. However, when analyzing network intrusion in general, we can suggest a different taxonomy:

- Signature-based detection of network intrusion and misuse.

While being very solid approach, with minimal possibility for a false-positive in case good signature database, signature-based detection requires frequent updates of the threat database and also substantial efforts in maintaining signatures. Exactly because of these significant efforts, there are a lot
of databases which are shared and that also increases the chance for a dedicated attacker to test his own malicious activity in advance against these databases and implements exactly the tweaks necessary to go undetected. Signature-based systems can be extended to apply fuzzy-logic ([4], [5]) and machine learning in order to improve their flexibility and responsiveness to new kind of cyber-attacks.

- **Anomaly-based detection algorithms.**

  With sufficiently large amount of network traffic to analyze, it is possible to detect patterns in normal user behavior and detect statistically significant deviations from it. The success of such approach depends on the quality of the patterns being detected/defined and on the predefined deviations boundaries which are considered to be “normal”. Taking into account that networks change over time and that user activities are also different, it takes a considerable amount of input data and a careful monitoring of the rules to have a highly accurate anomaly detection. Anomaly-based detection systems can also be combined with flexible AI tools like in [6], [7], [8], [9].

- **Hybrid detection algorithms.**

  The major goal of combining different detection algorithms is to try and detect both well-known as well as new threats. There could be different strategies toward mixing intrusions detection approaches. In this paper we focus mainly on use of machine learning to counter adaptable and AI-base cyber security threats.

### 2. Hybrid deep-learning model for anomaly detection

Hybrid systems can be built in a different way and for quite different purpose. It is often possible to get similar detection and discovery rates but with dissimilar performance ([9]). When choosing appropriate combination, the goal is to have it working as efficient as possible in both terms of accuracy and resources consumed (e.g. that it can handle as much traffic as possible with the provided hardware resources). Yet it is often necessary to choose between accuracy and versatility, which means that the most-accurate methods may not be suitable in terms of time required to analyze the data or in terms or necessary inputs. In this paper we follow the idea by Wang and Jones in [10] that solutions can be compared in terms of their flexibility, reliability, scalability, robustness and speed. Hybrid solutions typically score higher in the first four categories, compared to statistical and signature-based methods. This comes at the expense of being more complex, thus also harder to maintain and explain, and also requiring more resources and time to analyze the network activities.

We suggest a two-step procedure in building a hybrid anomaly detection. Layering different activities can improve flexibility as each layer can be replaced independently to build pluggable detection systems.

- The first step consists of feature extraction of the underlying traffic, which is adaptable but once executed the feature set remains fixed, until the entire system is reconfigured and re-trained. This makes it possible to remain flexible, while most important features are extracted once and then used. This reduces total calculation time and keeps resource usage low.

  Feature extraction step is shown on Figure 1, where traffic is captured with Wireshark and relevant packets are converted to XML or JSON for further analysis. It should be noted that filtering of packets based on specific protocols and ports is not included in the feature extraction – it can be used to simplify the initial step. The goal of feature extraction is to figure out which packet characteristics are most relevant to be included in the subsequent analysis (for example length, hosts etc.).

- The second step consists of classification algorithm that is responsible for giving hints if analyzed condition should be considered normal or not. This does not mean that output should be only a binary value – different classes of severity can be specified, or the output could be a probability for non-normal situation.

Once feature extraction has been complete, the remaining traffic captured is automatically filtered in accordance with the chosen feature set – e.g. for every packet captured, only features found to be relevant are processed. In case it is necessary to include new features, we can execute again model
calibration and run again the initial step. Although this is clearly a limitation for the suggested approach, regularly executing the initial step is also beneficial – it does not have to be done in real time, thus it has no impact on the performance of the whole system. On the other hand – tracking the differences in the important feature subset is a possibility to detect that there has been a change in network use.

**Figure 1.** Feature extraction and traffic analysis

For comparing the efficiency of the second step in suggested sequence, we use three different models – generalized linear model, support vector machine and neural network with BFGS optimization.

2.1. **Generalized linear model**

With a network under attack we assume that there will be no errors of our model that follow normal distribution. Therefore GLM ([11]) can be used to capture these effects, provided that we have a link function such that classification output (probability and/or class) Y, depends on our input characteristics X:

\[ E(Y) = f^{-1}(X\beta) \]  

(1)

In the simplest case where we want to test if the condition being analyzed is normal or not, we assume binomial distribution and use logistic function.

2.2. **Support vector machine**

Support vector machine ([12]) is used to find the maximum-margin hyperplane between the normal and a-normal cases of traffic. In our case, training vectors contain all features that have been extracted at the first step. Training of the algorithm is done over a training set of inputs, each containing a vector of characteristics and class of the situation (normal or not) \((x_{1i}, \ldots, x_{ki}, y_i)\). To allow for non-linear classification a hyperbolic tangent function kernel where \(\bar{x}_i = (x_{1i}, \ldots, x_{ki})\) and \(f(x_{i}', \bar{x}_i') = \tanh(\alpha(\bar{x}_i, \bar{x}_i') + c)\). It should be noted that SVM can be used also for novelty and outlier detection, but this is beyond the scope of this paper.

2.3. **Artificial neural network**
We use a simple neural network with a single layer of hidden neurons. The number of neurons inside is optimized through iterative procedure in a way that maximizes the accuracy and in particular the number of false positives. We focus on false positives as these results would likely lead to higher load for personnel responsible for incident checking and verifying if there has been indeed abnormal activity. Network used for processing packet data has a single hidden layer and 24 neurons inside.

3. Application and results
Data used to test the model consists of 257,673 packets and contains samples of ten different attacks, like denial of service, worms and backdoors. Unlike normal network traffic the amount of abnormal activity is high (39.3% of all cases) and this has been done in purpose in order to keep the test data set small.

Data preprocessing step is kept very simple on purpose, to allow 13 out of 45 network traffic parameters to be included and passed for processing for the second layer. Training set consists of 68% of all cases and the remaining data is used for assessing accuracy of models.

![Figure 2. ROC curves for GLM (top left), SVM (top left), ANN (bottom left) and hybrid model (bottom right)](image-url)

Hybrid model relies on the output of the three models but it is processed again through a simple neural network with single hidden layer containing 3 neurons. This makes it possible to create a dynamic ensemble model, where weights are also modified based on the performance of each
participating model. Adding more layers or more neurons is likely to increase the overfitting and actually reduce the accuracy of the combined model use.

Table 1. Accuracy in binary prediction of the test sample

| Model                              | Accuracy in classification (binary) |
|------------------------------------|-------------------------------------|
| Generalized linear model           | 0.881334                            |
| Support vector machine             | 0.808219                            |
| Artificial neural network          | 0.838144                            |
| Hybrid model using all three outputs | 0.909051                           |

Hybrid model can be extended further by plugging in specific rules that depend on the network topology and services exposed by different nodes. There are no such assumptions made with the current data set, but context-dependent rules can significantly reduce the false-positives and make sure that overall accuracy is improved. Artificial neural networks can be supported by other methods for tracking network activity, for example mean-reverting process modelling in which case normal load is considered to be a starting point and abnormal activities can be checked against their mean-reversion speed [13].

4. Conclusion

Machine learning algorithms can improve accuracy of intrusion detection and checking for anomalies in network activity. We tested several algorithms and found that hybrid methods, supported by additional rules can provide a competitive advantage and increase the overall detection accuracy, while at the same time decrease the amount of false-positives. Taking into account that network diversity and use increases it is especially important to keep track not only at the total correctness, but also at the efforts needed to track cases, which only seem like intrusion but are just normal communication.

Maximum accuracy achieved in the test case is a bit over 90%, which leaves further room for improvement, especially taking into account that means of network intrusion and cyber threats evolve over time. In this context, a major advantage of used detection methods is that they can be adapted and re-trained continuously to respond to changes in the environment. In this paper, out of 45 available features of network traffic only 13 are used, but this set can be extended and more sophisticated methods can be applied in the preliminary feature extraction. Combined with a set of extra rules and context-driven filters, that would significantly improve the detection rate and make it possible to automate further detection of network intrusion and cyber threats.

The research is partially supported by the NPD-Plovdiv University under grant No. FP19-FESS-014 “Contemporary tools and approaches for economic and business analysis”.

References

[1] Accenture, "Innivate for Cyber Resilience: Lessons from leaders to master cybersecurity," AccentureSecurity, 2020.
[2] G. Belani, "5 Cybersecurity Threats to Be Aware of in 2020," IEEE Computer Society, 2020.
[3] M. H. Bhuyan, D. K. Bhattacharyya and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," IEEE COMMUNICATIONS SURVEYS & TUTORIALS., vol. 16, no. 1, pp. 303-336, 2014.
[4] L. Ondrej, M. Manic, T. Vollmer and J. Wright, "Fuzzy logic based anomaly detection for embedded network security cyber sensor.," in IEEE Symposium on Computational Intelligence in Cyber Security, 2011.

[5] L. Wei, L. Ci and L. LiPing, "A New Method of Fuzzy Support Vector Machine Algorithm for Intrusion Detection," Applied Sciences, vol. 10, no. 3, p. 1065, 2020.

[6] F. Cheng, T. Li and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks.," in 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2017.

[7] A. Kumar, A. Abdelhadi and C. Clancy, "Novel anomaly detection and classification schemes for Machine-to-Machine uplink.," in IEEE International Conference on Big Data (Big Data), IEEE 2018, 2018.

[8] Y. K. Sharma and M. D. Rokade, "Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic.," IOSR Journal of Engineering, pp. 63-67, 2019.

[9] B. Belchev, A. Veleva and M. Ruseva, "Specialized software product "Statistical Analysis"," in International conference "Research Methods and Techniques in Economics and Social Sciences”, Plovdiv, 2012.

[10] S. Otoum, B. Kantarci and H. T. Mouftah, "On the Feasibility of Deep Learning in Sensor Network Intrusion Detection," IEEE Networking Letters, vol. 1, no. 2, pp. 68 - 71, 2019.

[11] L. Wang and R. Jones, "Big Data Analytics for Network Intrusion Detection: A Survey," International Journal of Networks and Communications, vol. 7, no. 1, pp. 24-31, 2017.

[12] A. J. Nelder and W. R. Wedderburn, "Generalized linear models.," Journal of the Royal Statistical Society: Series A (General), vol. 135, no. 3, pp. 370-384, 1972.

[13] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, p. 273–297, 1995.

[14] S. Kabaivanov and V. Markovska, "Modelling environment changes for pricing weather derivatives," Scientific Annals of Economics and Business, vol. 64, no. 4, pp. 423-430, 2017.

[15] A. Adriano, R. Conforti, M. Dumas, M. La Rosa and G. Bruno, "Automated discovery of structured process models: Discover structured vs. discover and structure.," International Conference on Conceptual Modeling, pp. 313-329, 2016.

[16] B. F. van Dongen, "BPI Challenge 2018," 4TU.ResearchData, 2018, 2018.

[17] G. Jansenswillen, B. Depaire, M. Swennen, M. Jans and K. Vanhoof, "bupaR: Enabling reproducible business process analysis," Knowledge-Based Systems, vol. 163, pp. 927-930, 2019.