Integrating Anomaly Detection for Enhanced Data Protection in Cloud-Based Applications

Konrad Czerkas[2], **Michał Drozd**[2[0009−0002−4577−3050]], **Agnieszka Duraj**[1], **Krzysztof Lichy**[1], **Piotr Lipiński**[1], **Michał Morawski**[1], **Piotr Napieralski**[1], **Dariusz Puchała**[1], **Marcin Kwapisz**[1], **Adrian Warcholiński**[1], **Michał Karbowańczyk**[1], **Piotr Wosiak**[1]

[1]*Lodz University of Technology*
Institute of Information Technology
Politechniki 8, 93-590 Łódź, Poland
piotr.napieralski@p.lodz.pl
[2]*LTC Sp. z o.o.*
Institute of Computer Science
Narutowicza 2, 98-300 Wieluń, Poland
kczerkas@finn.pl

DOI:10.34658/9788366741928.27

Abstract. *In this research, anomaly detection techniques and artificial neural networks were employed to address the issue of attacks on cluster computing systems. The study investigated the detection of Distributed Denial of Service (DDoS) and Partition attacks by monitoring metrics such as network latency, data transfer rate, and number of connections. Additionally, outlier detection algorithms, namely Local Outlier Factor (LOF) and COF, as well as ARIMA and SHESD models were tested for anomaly detection. Two types of neural network architectures, multi-layer perceptron (MLP) and recursive LSTM networks, were used to detect attacks by classifying events as "attack" or "no attack". The study underscores the importance of implementing proactive security measures to protect cluster computing systems from cyber threats.*
Keywords: *computer games, artificial intelligence*

1. Introduction

Cluster computing is a popular way to increase the computing power of systems. However, with this increase in power comes an increase in vulnerability to attacks. Two of the most common attacks on clusters are Distributed Denial of Service (DDoS) and Partition attacks [1, 2]. In order to detect these attacks, a series of experiments were conducted, where metrics were collected during normal

operation and then during anomalous behavior caused by external factors. The experiments showed that a solution was developed to detect DDoS and Partition attacks on clusters. The solution involved monitoring several key metrics, such as the number of connections, data transfer rate, and network latency. When these metrics exceeded a certain threshold, an alarm was triggered to alert the system administrator. The solution was able to accurately detect anomalies and distinguish between normal and anomalous behavior [3].

2. Anomaly Detection Techniques for Cluster Security

According to the commonly accepted definition of an outlier, it is a result of an observation that significantly differs from other results in a group. This difference suggests that this result is due to a different mechanism of generation. In this study, attacks were treated as anomalies or exceptions. The analysis of the stream began with determining the inner and outer lower and upper fences according to Tukey's method [4]. The extreme value exceeding the distribution limits was adopted as the definition of an outlier.

Two most popular algorithms for detecting outliers were also examined: the Local Outlier Factor[5](LOF) algorithm and COF. They define outliers based on the calculated coefficient. LOF creates so-called uniqueness ranks, while COF determines the isolation coefficient. It is assumed that an object (point) for which the LOF coefficient is approximately 1, e.g., $LOF \in (0.8; 1.2)$ belongs to the designated group of objects (belongs to the cluster). Objects for which the LOF coefficient changes abruptly relative to their local neighbors (upward and downward jumps can be observed) are called local objects (points) – local detected outliers. The COF isolation coefficient, on the other hand, determines how strongly a given object is isolated from the entire set. Two cases are considered, namely: the smaller the outlier index WW, the more objects the COF algorithm may indicate as outliers. If the outlier index equals 1, outliers will be those objects for which the isolation index is > 1. Two models directly related to anomaly detection in time series were also taken into account, namely the ARIMA (AutoRegressive Integrated Moving Average) model [6] and the Seasonal Hybrid Extreme Studentized Deviation (SHESD) model. SHESD detects one or more outliers in one-dimensional data streams that are approximately normally distributed. A necessary condition for detecting an outlier is to determine the upper limit of the predicted value of the given deviation. Unlike simulation studies, the experiments were conducted on a real cluster, which allowed for a reliable determination of the impact of request types on the collected load statistics. The cluster was built based on the OpenShift and Kubernetes systems, using an IBM rack server consisting of nodes. Instead, different types of loads were generated to observe the behavior of the cluster. The performance was also tested for the impact of the load balancer on response times

and error rates. Loads were generated from two computers, one of which simultaneously ran the DNS server and load balancer required for the cluster to function properly. Each computer had 6 CPU cores and 16 GB of RAM. The developed system detects anomalies to discover insider and outsider attacks from cloud centres. The proposed system has been evaluated using different datasets and its performance has been compared with several anomaly detection methods to determine its effectiveness when deployed on cloud data servers. The aim of the experimental research carried out was to test the effectiveness of neural networks of two selected types in detecting exceptions based on real data representing network traffic. Three parameters were selected as input data, representing the number of bytes in the system input, the average processor load per computational unit and the average response time per request, respectively. In addition, in order to determine the norms, data from consecutive days of individual months, which consequently form a yearly overview, were obtained and appropriately prepared, allowing good determination of daily, monthly or annual trends. Thus, properly analysed and processed data will constitute benchmark data which, in the best possible statistical sense, describe the behaviour of the system under normal operation. They thus constitute the 'ground-truth' for intelligent exception detection methods, i.e. situations that deviate from the norm, i.e. the normal observed operation of the system.

3. Artificial neural networks

Artificial neural networks are successfully applied to the task of attacks and anomalous behavior detection in computer systems and networks[7, 8]. For this reason, as the part of our research, we also considered artificial neural networks, focusing on the following two network architectures: multi-layer perceptron (MLP) and recursive LSTM networks. The task of attack detection was carried out in two schemes and both network architectures were used in both cases. In the first case, it was a problem of classifying events as "attack" or "no attack". The vectors of parameters (previously discussed) were fed to the input of a network, with the adjustable history window of size $L \in \{1, 2, 4, \ldots, 64\}$, and at the output of the network, we demanded a binary response indicating moments in time when an attack or an exceptional situation took place, which was further compared to the reference signal. This allowed to train neural networks in a supervised learning scheme using gradient techniques. In the second case, the task of attack detection was formulated as the prediction of parameters at a given time based on the history window ($L \in \{1, 2, 4, \ldots, 64\}$). Then depending on the accuracy of prediction and the specified threshold, it was decided whether the attack took place or not. Also here, both types of network architectures were tested. Based on the obtained results, we can conclude that in both considered cases, MLP and LSTM networks

enabled effective attack detection (understood as the distinct indication of a period of time when the attack took place). It should be noted that the predictive approach based on LSTM network allowed to detect attacks in the case of noisy data, when the signal-to-noise ratio was only 3dB.

4. Method

The proposed method for recognizing the current system state involves using a trained neural network model and a Python script that applies median filtering to the network's output. The input data is loaded from a CSV file containing the last 128 input data samples, which are then normalized based on coefficients and mean values obtained during the training process. The data is then prepared in packages of K samples, where K is a configurable parameter set in the INI file.

The trained model is loaded and applied to the prepared data, and the resulting outputs are compared with expected output vectors, which serve as reference points for decision making. The Euclidean distance between the actual output and each reference point is calculated, and the smallest distance indicates the system's current state. A median filter is then applied to the output to smooth out any dynamic changes, and a decision is made based on the final output.

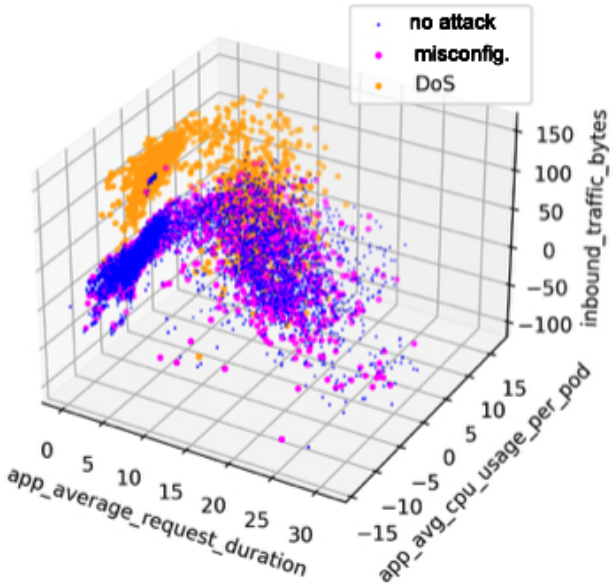
The following equation can summarize the method:

$$wdata = \text{fmedian}(\arg\min_{j \in 1,\ldots,4} \|z - vv_j\|) \tag{1}$$

where *wdata* is the final output, z is the output of the trained neural network model, *vv* is the expected output vector, and $fmedian$ is the median filter function.

Analysis of the results allows important conclusions to be drawn. Certainly, the network tends to activate in discrete moments where the attack has not taken place. Likewise, discrete moments indicating the absence of an attack may occur when an attack has occurred. Adding a median filter helps to smooth out such dynamic changes resulting in results closer to the expected ones. We can base the above intuition on the assumption that the attack lasts for a longer period. The results obtained from the neural network-based intrusion detection system showed promising performance in accurately detecting and classifying network attacks. The use of a median filter to reduce the impact of dynamic changes in the input data resulted in more stable and accurate outputs. The confusion matrix showed that the model achieved high accuracy in classifying different types of attacks, with only a small percentage of misclassifications. Overall, the developed system has the potential to provide reliable and effective network security for various types of organizations.

Values on the main diagonal indicate correct recognition. Values off the main diagonal represent misrecognition results. The sum of the elements lying outside the main diagonal is 7.9%. This is a good result.



	No Attack	DoS	Misconfiguration	Both Attacks
No Attack (predicted)	61.05%	1.3%	1.8%	0.2%
DoS (predicted)	0.9%	14.7%	0.2%	0.5%
Misconfiguration (predicted)	1.9%	0.2%	13.0%	0.2%
Both Attacks (predicted)	0.2%	0.4%	0.1%	2.8%

Figure 1. Visualisation of input data as a point cloud and Results as a confusion matrix. Source: own work.

It should be noted that the task of classifying the input data into the four classes considered is not a trivial issue. The complexity of the problem is best demonstrated by visualising the input data as a point cloud in three-dimensional space.

The analysis of the results suggests that the network tends to activate in discrete moments where no attack occurred, and conversely, may have moments indicating the absence of an attack during actual attacks. Using a median filter helps to alleviate these dynamic changes, leading to more accurate results. This can be attributed to the assumption that attacks usually last for a longer period of time. In conclusion, the study suggests that adding a median filter can improve the accuracy of IDS systems in detecting attacks.

5. Conclusions

The focus of this research is to emphasize the significance of identifying and mitigating attacks on cluster computing systems. The development of a solution

to detect Distributed Denial of Service (DDoS) and Partition attacks is a proactive approach for system administrators to safeguard their systems and avoid any possible downtime. The metrics utilized in this research can be utilized as a foundation for creating more advanced security solutions for cluster computing. This study underlines the need for continuous improvement and implementation of security measures to protect cluster computing systems from various cyber threats.

Acknowledgment

The research presented in this article was made possible through the financial support of "Środowisko budowy i eksploatacji bezpiecznych aplikacji działających w chmurze w oparciu o inteligentne wykrywanie anomalii w klastrach obliczeniowych oraz techniki kryptograficzne blockchain/DLT (CL) Nr Umowy z NCBR: POIR.01.01.01-00-0263/21-00".

References

[1] Nugraha B., Kulkarni N., Gopikrishnan A., *Detecting adversarial ddos attacks in software- defined networking using deep learning techniques and adversarial training*, [In:] *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, pp. 448–454, doi: 10.1109/CSR51186.2021.9527967.

[2] Wang B., Zheng Y., Lou W., Hou Y.T., *DDoS attack protection in the era of cloud computing and software-defined networking*, *Computer Networks*, 2015, vol. 81, pp. 308–319, ISSN 1389-1286, doi: https://doi.org/10.1016/j.comnet.2015.02.026.

[3] Yan Q., Yu F.R., Gong Q., Li J., *Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges*, *IEEE Communications Surveys & Tutorials*, 2016, vol. 18, no 1, pp. 602–622, doi: 10.1109/COMST.2015.2487361.

[4] Tukey J.W., *Comparing individual means in the analysis of variance*, 1949, vol. 5, pp. 99–114, ISSN 0006-341X (print), 1541-0420 (electronic), doi: https://doi.org/10.2307/3001913.

[5] Breunig M.M., Kriegel H.P., Ng R.T., Sander J., *Lof: Identifying density-based local outliers*, *SIGMOD Rec.*, 2000, vol. 29, no 2, p. 93–104, ISSN 0163-5808.

[6] Alam T., *Predicting revenues and expenditures using artificial neural network and autoregressive integrated moving average*, [In:] *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, pp. 1–4.

[7] de Campos Souza P.V., Guimarães A.J., Rezende T.S., Silva Araujo V.J., Araujo V.S., *Detection of anomalies in large-scale cyberattacks using fuzzy neural networks*, *AI*, 2020, vol. 1, no 1, pp. 92–116, ISSN 2673-2688.

[8] Bongiovanni W., Guelfi A.E., Pontes E., Silva A.A.A., Zhou F., Kofuji S.T., *Viterbi algorithm for detecting ddos attacks*, [In:] *2015 IEEE 40th Conference on Local Computer Networks (LCN)*, pp. 209–212, doi: 10.1109/LCN.2015.7366308.