Cascaded Anomaly Detection with Coarse Sampling in Distributed Systems

Amelia Bădică[3], Costin Bădică[3], Marek Bolanowski[1(✉)], Stefka Fidanova[4], Maria Ganzha[2], Stanislav Harizanov[4], Mirjana Ivanovic[5], Ivan Lirkov[4], Marcin Paprzycki[2], Andrzej Paszkiewicz[1], and Kacper Tomczyk[1]

[1] Rzeszów University of Technology, Rzeszów, Poland
{marekb,andrzejp}@prz.edu.pl
[2] Polish Academy of Sciences, Warszawa, Poland
[3] University of Craiova, Craiova, Romania
[4] Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia, Bulgaria
[5] University of Novi Sad, Faculty of Sciences, Novi Sad, Serbia

Abstract. In this contribution, analysis of usefulness of selected parameters of a distributed information system, for early detection of anomalies in its operation, is considered. Use of statistical analysis, or machine learning (ML), can result in high computational complexity and requirement to transfer large amount of data from the monitored system's elements. This enforces monitoring of only major components (e.g., access link, key machine components, filtering of selected traffic parameters). To overcome this limitation, a model in which an arbitrary number of elements could be monitored, using microservices, is proposed. For this purpose, it is necessary to determine the sampling threshold value and the influence of sampling coarseness on the quality of anomaly detection. To validate the proposed approach, the ST4000DM000 (Disk failure) and CICIDS2017 (DDoS) datasets were used, to study effects of limiting the number of parameters and the sampling rate reduction on the detection performance of selected classic ML algorithms. Moreover, an example of microservice architecture for coarse network anomaly detection for a network node is presented.

Keywords: Anomaly detection · Anomaly prediction · Complex distributed system · Computer network management

1 Introduction

Despite improving quality of devices used in modern computer systems, failures continue to occur. Anomalous functioning can be caused by a hardware failure or be the result of an external factor, such as an attack. Not detecting the anomaly on time may result serious consequences, including large financial losses. Hence, failure prediction may reduce damages and costs of maintenance and repair. Besides, it may reduce system downtime and ensure service reliability for the end users. Anomaly detection is, typically, based on analysis of selected system parameters, during normal operation and

S. Sachdeva et al. (Eds.): BDA 2021, LNCS 13167, pp. 181–200, 2022.
https://doi.org/10.1007/978-3-030-96600-3_13

during failures. It is a well-known fact that in production systems, occurrence of an anomaly can be conceptualized mainly in two scenarios. (1) Anomaly detection subsystem informs about an adverse event that is "in progress", or that has ended, and the goal of administrator(s) is to limit impact of its (possible/future) effects on system's operation (e.g., DDoS detection). (2) Anomaly prediction informs of the deteriorating operational parameters, suggesting the system failure. Here, it may be possible to react in time to avoid occurrence of an actual failure (e.g., hard disk drive crash). However, in what follows, unless explicitly stated, anomaly detection and anomaly prediction are treated similarly, and terms are used interchangeably. This is because the focus of this work is on the detection itself, rather than its context.

Majority of research on anomaly detection is based on application of Machine Learning (ML, hereafter) techniques. Here, ML is applied to data collected within the observed systems. However, as it will be shown in Sect. 3, due to the amount of generated data, in most cases, only key components of large systems can be monitored. This is why, a novel method for system monitoring and anomaly detection (cascading anomaly detection with coarse sampling) has been proposed. It allows efficient monitoring of a substantially larger number of system components, facilitates distributed probing, and reduces the overall size of the analyzed dataset.

In this context, we proceed as follows. Section 2 contains summary of the state of the art. While it covers a relatively broad area of anomaly detection research, this contribution is focused on anomaly prediction in distributed computer systems. Architecture of the proposed approach is discussed in Sect. 3. Particularly important for the proposed approach are effects of limiting sampling frequency on the effectiveness of fault detection (considered in Sect. 4). This is particularly important in the case of IoT ecosystems, which are often characterized by limited bandwidth of communication channels. Section 5 summarizes the results, indicating fields of their potential implementation, and directions of future research.

2 Related Literature

Due to the scale of distributed systems, it is practically impossible to entirely rely on human-based management. Instead, automatic recognition and prediction of anomalies has to be applied [1]. For the failure prediction, algorithms based on supervised learning are mainly used. Here, in recent years, deep learning-based approaches received considerable attention [2]. Many studies have noted that certain anomalies manifest themselves as deviations of specific performance parameters. In this context, Williams et al. [3] studied which performance metrics are affected by different types of faults. These parameters included CPU usage, available memory, and network traffic, among others. On the basis of collected data, system can learn to distinguish between correct behavior and anomalies. Unfortunately, this solution does not provide information about the cause of failure and does not "confirm" failure occurrence. It only delivers "preliminary information" about possible problems. A similar approach can be found in [4], focusing on multilayer distributed systems. A solution, called PreMiSE, combines anomaly-based and signature-based techniques to identify failures. Here, it is possible to distinguish, which anomalous situations are "correct" and can lead to a failure. Based on historical data,

PreMiSE learns to associate the type of failure with specific malicious anomalies. In [5], recurrent neural networks were used to predict task failures in cloud systems, and prediction accuracy of 84% was reported. Work of Zhao et al. [6], attempted to address difficulty of describing large datasets from, for example, cloud data centers. To avoid this problem, the authors proposed a system based on k-nearest-neighbor (k-NN) clustering. Using principal component analysis [7], they extracted the most relevant features and reduced the dimension of the data matrix. Next, k-NN and the modular community detection were applied to find the largest cluster, which was assumed to represent the normal functioning, while other clusters were claimed to represent anomalies.

A large number of anomaly detection systems were applied to clouds, or to core transmission links. Here, increasing computational efficiency of detection systems allows for increasing the number of end devices and their components to be analyzed. One of the topics, emerging in recent research, is prediction of computer hardware failures [8]. Of course, due to variety of failures that may occur, ML algorithms cannot predict every failure. To use them effectively, it is necessary to select pertinent indicators that can inform about the symptoms that precede the actual failure of a selected class of components (e.g. database servers). One of the most important components, with a high probability of failure is the hard drive [9]. Exact prediction of time of failure of the selected hard drive is extremely difficult. Since, for all practical purposes, such precise prediction is not necessary, the concept of time window, indicating the approximate time of failure, was introduced. This allows early reaction, mitigating core dangers. One of the proposed approaches is to build a statistical model, consisting of several environmental and operational parameters [10]. Here, the influence of individual indicators is determined, and dependencies between them are identified. In this context, recurrent long-short term memory (LSTM) neural networks, applied for time series prediction, have been used in [11]. Here, authors managed to predict disk failures, in the next 15 days time-window, with an accuracy of 86.31%. On the other hand, system using XGBoost, LSTM and ensemble learning, achieved prediction accuracy of 78% for a 42 day window [12]. Bothe these results are reasonable for production systems.

Fault prediction materializes also in power systems, where faults lead to severe problems and power outages [13]. An approach to solving power disruption problem was proposed by Omran et al. [14]. Here, authors used various ML techniques to achieve the highest possible prediction accuracy. Their proposed system consists of data cleaning, conversion, and classification, using selected features. Applying an ant colony optimization, the most important features of the dataset are selected. Next, a number of classifiers, such as k-NN, Artificial Neural Networks, Decision Tree (DT), Logistic Regression, and Naïve Bayes (NB) were used. Performed experiments resulted in accuracy between 75% and 86%.

A different approach to the problem has been found in works dealing with power cable faults [15, 16]. As a solution, statistical analysis of data collected over a long time period, consisting of various parameters that describe network load, and environmental influence, was used.

Industrial Internet of Things (IoT) environments are another area where fault prediction is important. Due to the (very) large number of sensors used to check the status of machines and manufacturing processes, huge amount of data is collected.

Therefore, it is important to use an appropriate feature selection algorithm, to discard data that has no impact on fault prediction. Study by Kwon et al. [17], proposed a fault prediction model with an iterative feature selection. First, data is prepared by splitting, elimination, and normalization. Next, using random forest algorithm, the importance of each feature is established, based on its correlation with failure(s). Then, different sets of features are iteratively selected, and models are built on their basis, using the Support Vector Machine (SVM) classifier. Finally, model selection is performed on the basis of prediction accuracies. In their work, Fernandes et al. [18] focused on prediction of heating equipment failure. An IoT platform that provides sensor information was used as a data source. Supervised learning was applied to find classes that describe the fault condition. Feature selection, based on a random forest, was used to discard irrelevant attributes. The LSTM neural network was applied to the time series.

Another method, appearing in various publications, related to failure prediction, is the Bayesian network [19]. An example of its application is software fault prediction [20]. Besides, Bayesian networks have been used for anomaly detection, diagnosis, or time series prediction.

One more area, where anomaly detection has been applied, are computer networks [21–23]. Here, popular methods are based on mining error logs, and analysis of copies of network traffic samples. In the work of Zhong et al. [24], the investigation of data from alarm logs of a metropolitan area network, over a period of 14 months, was carried for classification of the state of the operation of the system (normal, abnormal, failure, and sudden failure). In [25], similarly as in above mentioned research, data is derived from logs. However, this time, a simulated wireless telecommunication system is considered. As a result of unstructured nature of the data, it had to be processed and labeled. A convolutional neural network model was used to make predictions about the state of the network.

3 Architecture of the Proposed System

Majority of the approaches described above are applied to monitoring only the key elements of a system. Here, a new architecture for anomaly detection and prediction in distributed systems, which can include majority of its functional elements, is proposed. To achieve this, cascade model, with coarse sampling, is to be used. A scheme of the model is shown in Fig. 1a. At each level (cascade), a set of different tools for anomaly detection is to be implemented, according to the following principle: the first cascade uses the least accurate methods with low computational complexity, capable of processing (very) large amount of data (very fast); in next cascade, level of expected accuracy of anomaly detection increases. However, here only events that have been "flagged" by the first level of the cascade are considered. Hence, the amount of data that is (has to be) taken into account is considerably smaller. The last cascade "decides" whether the received data indicates an occurrence of an actual anomaly. Overall, if an anomaly is detected as possible, at a given level, pertinent data is passed to the next cascade for further (more detailed; resource consuming) analysis. This approach allows monitoring a wider range of (non-key) elements of the system. Of course, data from critical elements (e.g., production servers) may be routed directly to the "most accurate" cascade.

Here, it is crucial to capture problems "as they materialize", without waiting for the cascading system to "capture them". In this contribution, a dual cascade system is considered. Moreover, the reported research is focused on optimizing the performance of the first cascade.

Separately, it has to be stressed that in anomaly detection, the key issue is to minimize the impact of the detection mechanisms on the functioning of individual elements of the system. To meet this requirement, additional probes may be installed. These constitute a subsystem independent from the monitored system, which can monitor, for instance, network core connections or Internet service provider access links, using specialized out-of-band intrusion detection (IDS) and intrusion prevention (IPS) subsystems. However, this approach has two major drawbacks: (1) it sends large amount of data to the centralized detection system(s), and (2) high performance of these systems needs to be guaranteed. Poor performance of IDS/IPS can introduce additional latency, which may be unacceptable. Here, work [26] proposes native mechanisms to copy traffic from selected switches and forward it for analysis outside of the production transmission path. This approach eliminates the problem of delays introduced by the anomaly detection process. It can also be successfully applied to monitor other components of a distributed system (not only network devices).

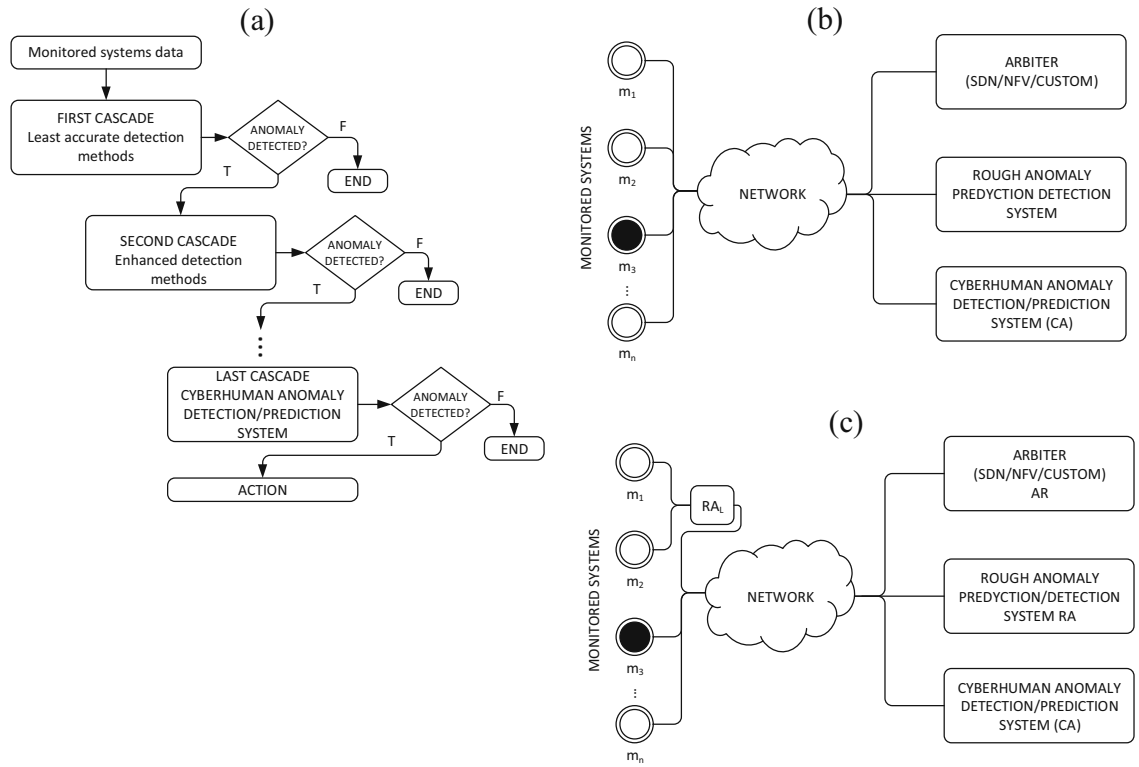

Fig. 1. **a.** Model of a cascaded anomaly detection system; **b.** Centralized coarse anomaly detection system; **c.** Distributed anomaly detection system.

In this context, the scheme of the centralized system with coarse anomaly detection is presented in Fig. 1b. The monitored systems $m_1, m_2 \ldots m_n$ (network devices, computer systems, printers, IoT components, etc.) send diagnostic information via a (possibly separate) network to the supervisory systems, i.e.:

- **Rough anomaly prediction/detection (RA) system** – which is responsible for analysis of large sets of diagnostic data, from all supervised devices. RA systems can be implemented in the form of hardware-software probes, which are inserted in selected places within the monitored system. The detection algorithms used in RAs should be characterized by lowest possible complexity, and a minimum acceptable detection threshold (configurable by the administrator). When a potential anomaly is detected, pertinent data is forwarded to the CA system.
- **Cyber-Human anomaly prediction/detection system (CA)** – is a collection of IDS and IPS systems, which uses highly accurate (though computationally complex) detection systems. Its results are to be verified by the engineering team. Note that in critical systems, such as production servers, it may be necessary to send full diagnostic data directly to the CA, bypassing the RA. Observe also, that in multi-step cascade systems, it is possible to instantiate (in selected locations) additional RA layers, with "intermediate level of sensitivity" of anomaly detection.
- **Arbiter (AR)** – system for control and management of monitored devices, which, based on data from the anomaly detection systems, determines parameters of the monitored system and an identified event. On this basis, it may undertake reconfiguration actions, e.g.: disabling a device, blocking it using access control lists, reconfiguring routing, sending a service request, performing a backup, etc. The AR architecture can be built based on the SDN (software-defined networking) model, NFV (network functions virtualization), Ansible, or by using custom solutions written in Python, with the Netmiko and Paramiko libraries [27–29].

As noted, in case of large ecosystems, collecting data from all supervised devices puts a significant load on the network. The following approaches can be used to reduce the network load:

- Monitoring only key system elements (e.g. m_3, marked in black), is a common approach, which obscures the complete picture of the state of the system. As a side note observe that deficit of ICT engineers makes it increasingly difficult to ensure regular review of non-key IT systems (e.g. disks in office PCs). Hence, only by introducing automatic supervision of all (as many as possible) systems, it is possible to allow prediction of failures on non-core components and undertaking timely remedial actions.
- Relocating very coarse detection systems, e.g. RA_L (see, Fig. 1c), where the data is generated (e.g. in the three cascade approach). This reduces transfer of diagnostic data across the network. Data is sent to RA and AR only when potential anomalies are detected. The RA_Ls, operating within a single system, may differ in applied classifiers and their settings, to match specific classes of monitored devices. Their relatively simple structure allows them to be implemented as microservices, and deployed anywhere in the distributed system, e.g. on servers, IoT devices, gateways, PCs, etc. Moreover, RA_Ls, as the first cascade step, can be dedicated to non-key devices, where high accuracy of threat detection may not be required. This also reduces the complexity of the detection process (for such systems) and reduces the total number of RA_L probes in the system.

- Using an appropriate (reduced) sampling rate for diagnostic data sent to RA and/or RA_Ls, to reduce the data transfer. Moreover, using only "key parameters" to train the ML model (while omitting the less "important ones") can also lead to faster model training and application. Here, in both cases, it is necessary to determine what effect removal of some available data has on the accuracy of anomaly detection. This aspects of the proposed approach is considered in detail in Sect. 4.

4 Effect of Number of Parameters and Sampling Frequency on Model Accuracy

To establish relation between the number of parameters and the number of samples and the quality of anomaly detection, data from two, publicly available, datasets was used:

- ST4000DM000 dataset (from 2017) was selected from the Blackblaze website [31]. It contains statistics of hard drives from data centers. Each row of data refers to a single day, and contains, among others, the following attributes: time, disk serial number, model, capacity, and statistics derived from the S.M.A.R.T. protocol. Additional column, named "Failure" having value "0" if the disk is operational, or "1" if it is the last day before failure, is also present. Table 1 shows the number of records assigned to each class describing disk performance.
- CICIDS2017 [32] dataset contains 72 attributes of network traffic, during normal activities and during various types of attacks. Here, we have taken into consideration data related to a DDoS attack. Data was collected during five days, in the form of pcap's and event log files. Additional column, named "Label", having a value of "0" if there is no DDoS attack and "1" when packets are part of the DDoS attack, was also included (Table 2).

Table 1. Class types and record counts in dataset ST4000DM000

Normal ('0')	Failure ('1')	Total number of records
167823 (94,1%)	10477 (5,9%)	178300

Table 2. Class types and record counts in dataset CICIDS2017

Normal ('0')	Failure ('1')	Total number of records
77585 (26%)	221485 (74%)	299070

In this work, which should be treated as a preliminary step of a long-term project, four classic ML algorithms were experimented with: SVM, DT, k-NN and NB. In the future, we plan to look into pertinent modern approaches, e.g. encoders, LSTM and CNN neural networks, etc. Models were implemented in MATLAB, with the "Statistics and Machine

Learning Toolbox" add-on [30]. Standard versions of the following measures were used to compare model performance: accuracy, precision, recall, specificity, and F-measure. Before proceeding, let us recall (and stress) that at this junction we are interested in methods that are to be applied in the first cascade. Hence, they should be "relatively accurate", while consuming minimal amount of resources, and being fast in training and application.

4.1 Reducing Number of Attributes

In the first series of experiments, effects of reducing the number of attributes were studied. To determine the order of adding attributes, the MRMR (minimum redundancy maximum relevance) algorithm was used, which returns attributes in order of importance to the classification. Accordingly, attributes were added starting from these that were established to be the most relevant. Figure 2, Fig. 3, Fig. 4 and Fig. 5 show results of the DT, SVM, k-NN and NB, for the dataset ST4000DM000, for different number of attributes, added in the following order (we list the first 10): Current Pending Sector Count; Reported Uncorrectable Errors; Power-off Retract Count; Reallocated Sectors Count; Uncorrectable Sector Count; End-to-End error; UltraDMA CRC Error Count; Temperature; Total LBAs Read; Start/Stop Count.

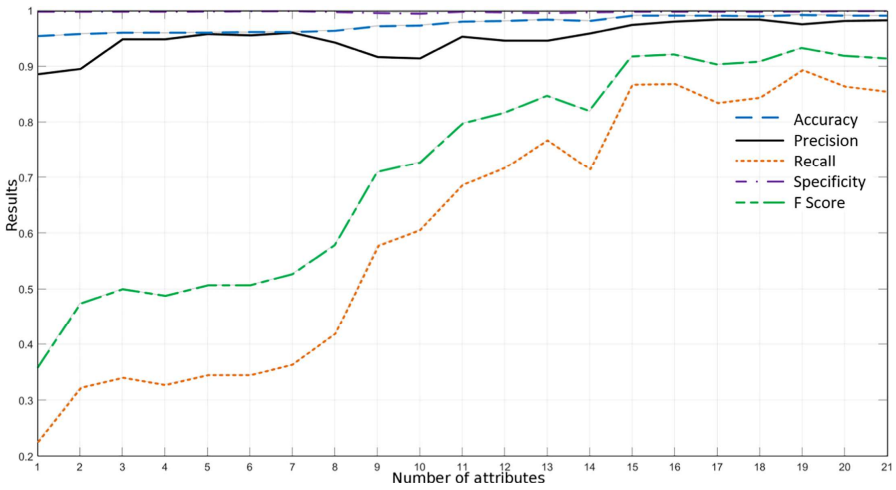

Fig. 2. Effect of varying the number of attributes for the DT model; ST4000DM000 dataset.

In case of DT (Fig. 2) particularly noticeable is the change in the recall measure responsible for the model's ability to detect abnormal disk behavior. It can be concluded that the DT model needs most of available attributes to get a satisfactory result. For the SVM algorithm, a similar relationship was observed (see Fig. 3). However, for this algorithm, addition of the last two attributes resulted in a significant decrease in both F-Score and Recall.

With the k-NN algorithm (Fig. 4), high sensitivity of performance, to successively added initial (most important, according to the MRMR) attributes was observed. As the number of attributes increased, the performance measures oscillated, until the number of attributes reached 15, where they stabilized at very high values. Interestingly, again, adding the last attribute had a negative impact on the results.

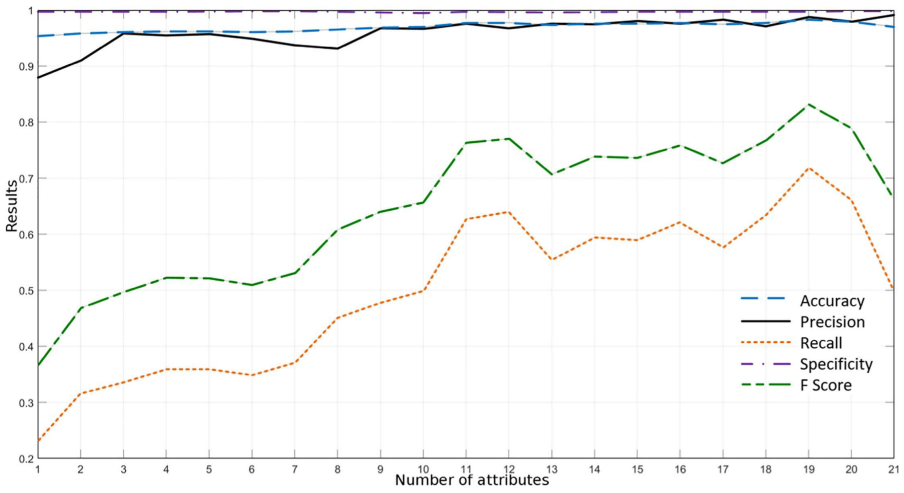

Fig. 3. Effect of varying the number of attributes on SVM model; ST4000DM000 dataset.

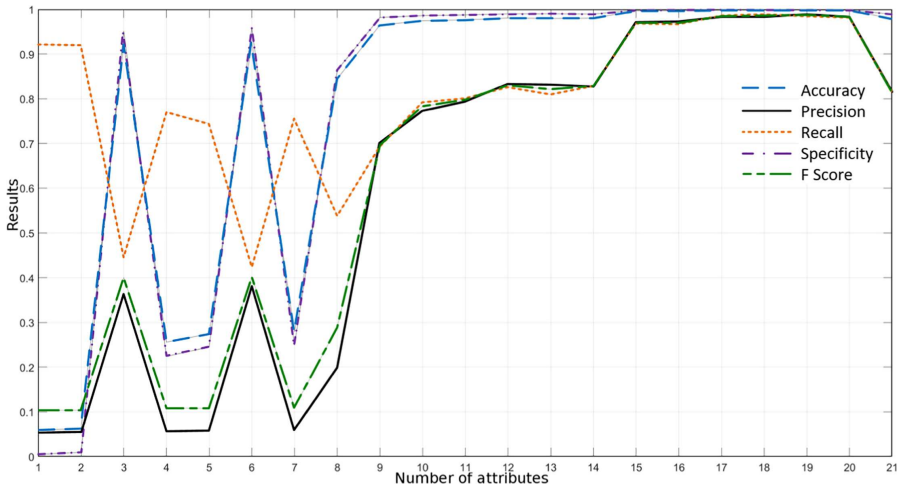

Fig. 4. Effect of varying the number of attributes on k-NN model; ST4000DM000 dataset.

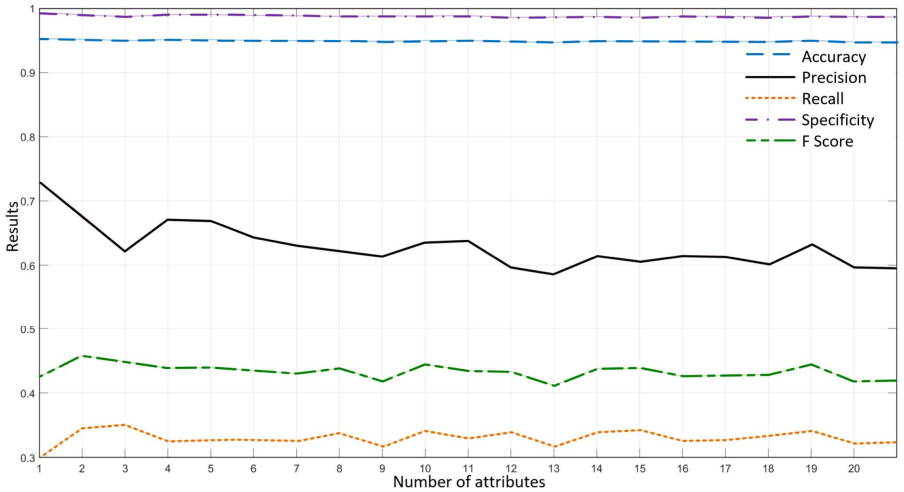

Fig. 5. Effect of varying the number of attributes on NB model; ST4000DM000 dataset.

The NB algorithm obtained poor results regardless of the number of attributes used (Fig. 5). The results show clearly that this approach is not suitable for this type of data.

Figure 6, Fig. 7, Fig. 8 and Fig. 9 show results for the CICIDS2017 dataset, for different number of attributes, added in the following order (we list the first 10): Init_Win_bytes_forward, Init_Win_bytes_backward, MinPacketLenght, ActiveStd, BwdPacketLenghtMean, FwdIATMin, FwdPacketLenghtMax, BwdPacketLenght-Mean, IdleStd, TotalLenghtOfBwdPackets.

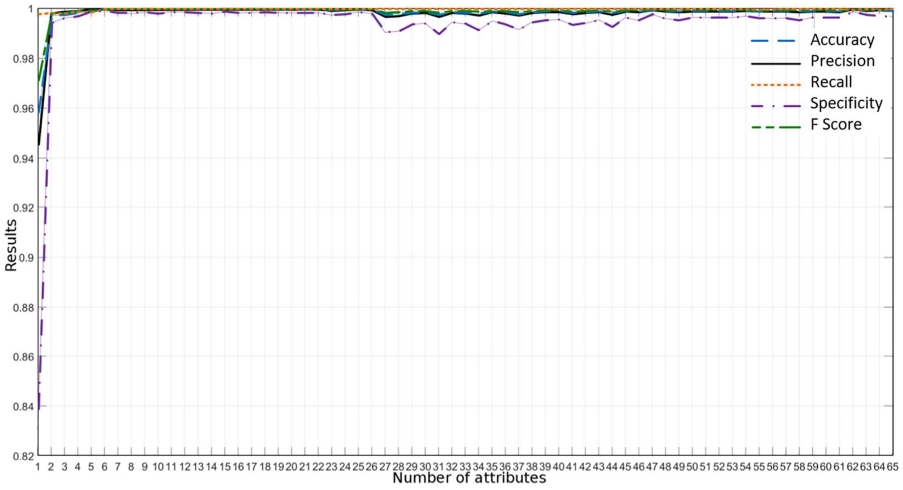

Fig. 6. Effect of varying the number of attributes on DT model; CICIDS2017 dataset.

For the DT algorithm (Fig. 6), results show high classification ability for as few as two of the most important attributes. All performance measures, for more than two parameters, obtained values greater than 98%. However, increase of the number of attributes resulted in a slight deterioration of the specificity measure. This would seem to suggest that DT, with 2–4 attributes should is a reasonable candidate for the first cascade step in the proposed approach to anomaly detection.

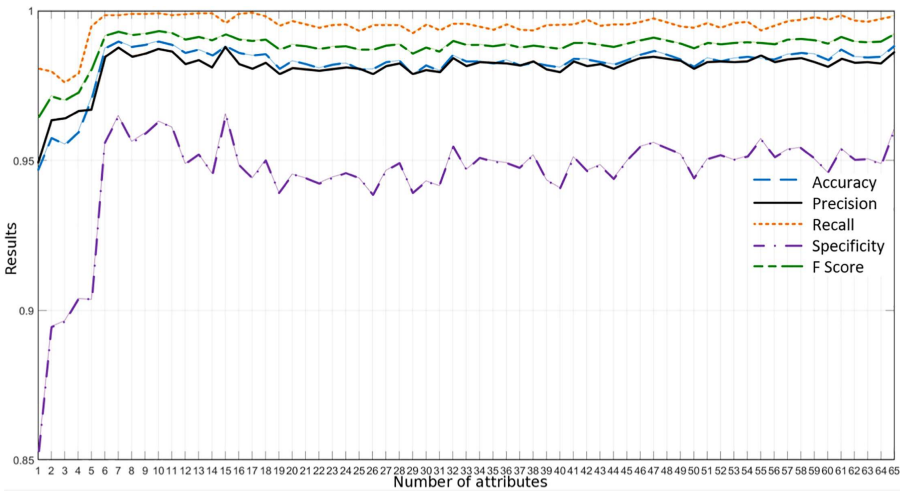

Fig. 7. Effect of varying the number of attributes on SVM model; CICIDS2017 dataset.

Figure 7 shows the result for the SVM algorithm. Interestingly, the best performance was obtained for 6–15 attributes. There exists some level of similarity to the results, obtained by the SVM, for the ST4000DM000 dataset. The results obtained for the Specificity measure remain very low regardless of the number of included parameters.

Figure 8 shows results of the model dependence on the number of attributes for k-NN algorithm and the CICIDS2017 dataset. Here, the results are rather peculiar as sudden performance jump can be observed after 35th parameter is added. From there on, very high performance of all measures is observed. It is unclear why this specific parameter brought this change. However, taking into account that the DT approach reaches high level of accuracy (for all measures) already for very few parameters, we have decided to not to investigate this peculiar effect further.

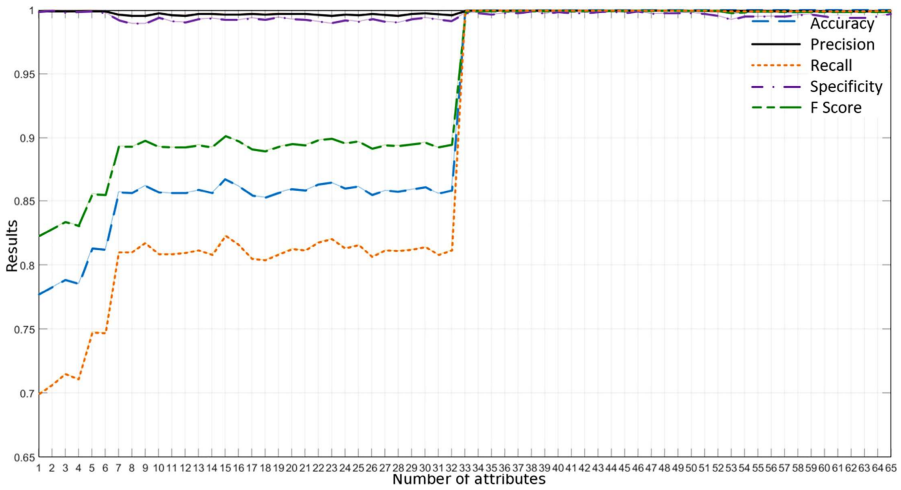

Fig. 8. Effect of varying the number of attributes on k-NN model; CICIDS2017 dataset.

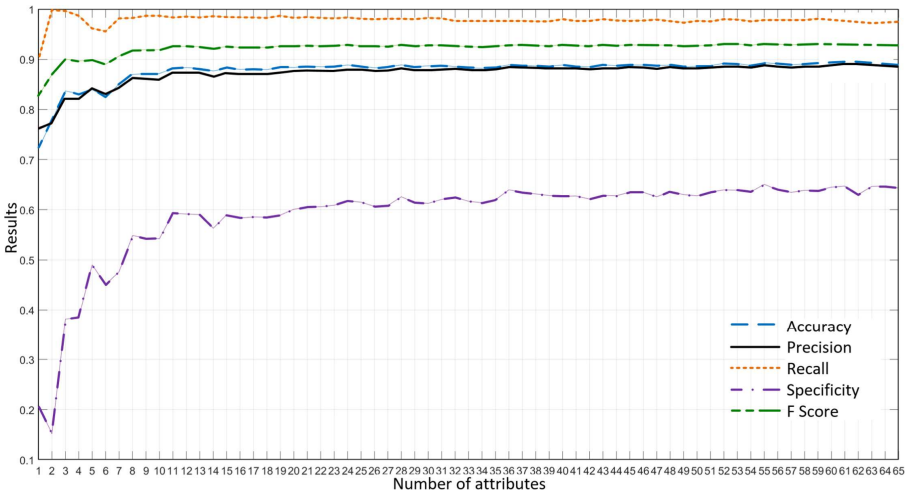

Fig. 9. Effect of varying the number of attributes on NB model; CICIDS2017 dataset.

Figure 9 shows the results for the NB algorithm and the CICIDS2017 dataset. The figure shows a significant increase in measure values at the initial stage of attribute addition. Already at about half of the added attributes, the models gains its maximum performance, which remains stable. However, again, also for this dataset, the performance of NB is substantially worse than that of the remaining three classic algorithms.

Overall, taking into account the results obtained when investigating effects of reducing the number of analyzed attributes; for the prediction of disk damage the best results were obtained after application of the k-NN algorithm, for 16–20 attributes. In the case of problems related to the detection of anomalies for network traffic, the best turned out to be the DT algorithm, which obtained very good results already for two attributes. These results indicate clearly, that anomaly detection, performed by the first step of the cascade may really need to use different ML approaches, when anomalies in different contexts are to be detected.

4.2 Reducing the Size of the Learning Set

In the second series of experiments, the effect of granularity of data sampling, on the performance of predictive models was explored. In these experiments, the same performance metrics were applied. This allowed us to observe how much the ML process can be accelerated at the cost of a small decrease in model accuracy.

Initially, it was decided to use a systematic sampling approach. In this approach, we select every $2n$th element in the entire set, where $n \in \{1, 2, \ldots, 13\}$. For example, for $n = 1$ elements with the following numbers: 1, 2, 4, 6, 8, ... were selected. However, for the ST4000DM000 dataset, a large class imbalance was observed, with the number records in class 1 being only 5.9% of the total set. This led to a situation where smaller data samples contained records belonging only to class 0, which made it impossible to achieve valid results. Therefore, simple random sampling of the dataset was used (to assure existence of both classes in the dataset. However, this issue remains open for the case of practical realization of the proposed approach. Not being able to collect enough data samples of Class 1 to initialize the model needs to be taken into account. Here, the potential of application of transfer learning (e.g. using the ST4000DM000 dataset) may need to be considered. In addition, it was also decided to use 10 times repetition of the test, to calculate the average value for the obtained measures. Figure 10 shows a comparison of results for systematic and random sampling for the DT algorithm on ST4000DM000 data.

Figure 11 through Fig. 16 show the results for the other algorithms applied on the ST4000DM000 and CICIDS2017 datasets.

For the ST4000DM000 dataset, and the DT algorithm, a large decrease was observed in precision, recall and F-measure, which are related to the model's ability to predict class 1, which could signal the possibility of disk failure. When the entire dataset was analyzed before sampling, the values of these parameters were close to 100%, while for 10% of the input data, the values dropped to about 78%.

For the CICIDS2017 dataset, the results for the DT algorithm are shown in Fig. 12. Here, a fairly slow decrease in the performance measures can be observed. The largest drop was observed for the specificity measure.

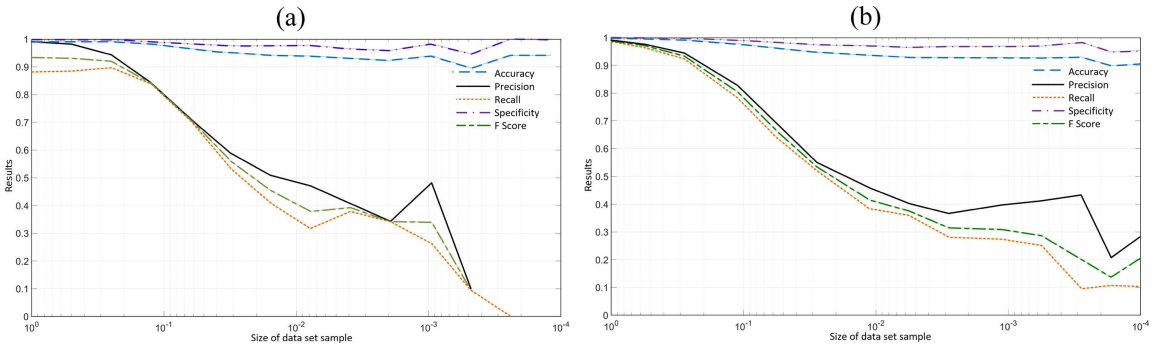

Fig. 10. Comparison of systematic (a) and random (b) sampling for the ST4000DM000 dataset for the DT algorithm.

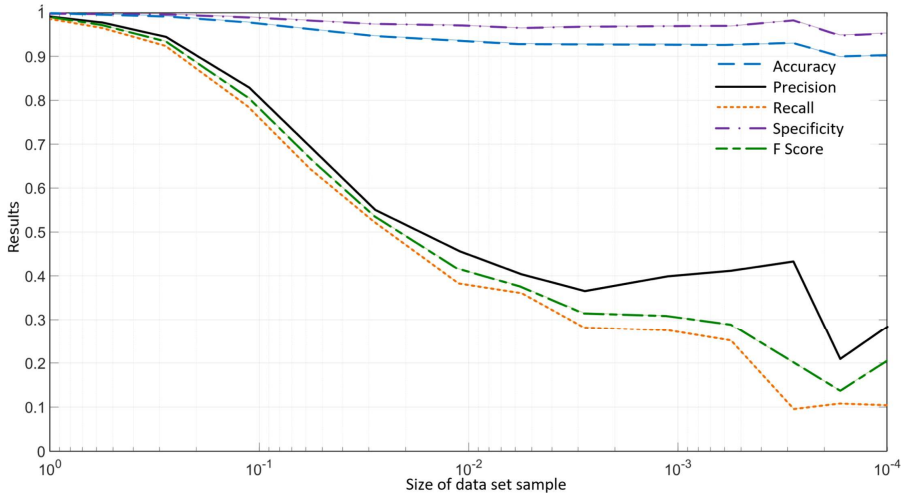

Fig. 11. Effect of changing the number of records on DT model; ST4000DM000 dataset.

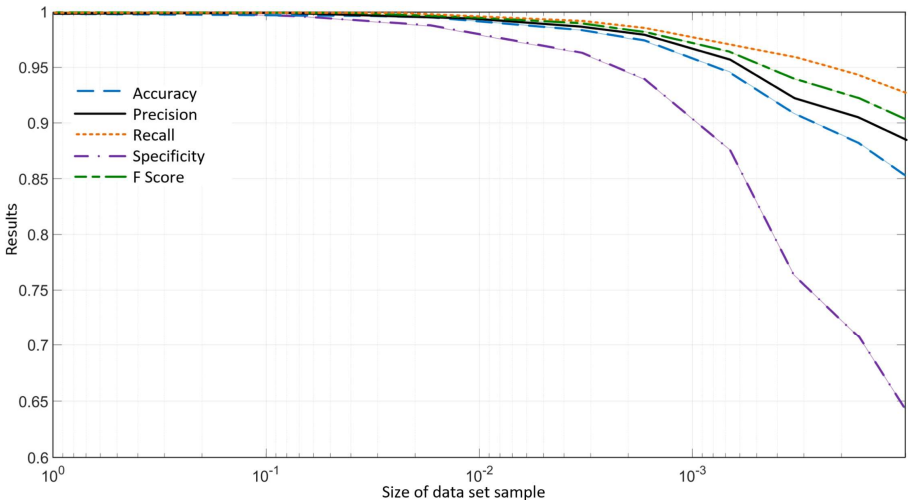

Fig. 12. Effect of changing the number of records on the DT model; CICIDS2017 dataset.

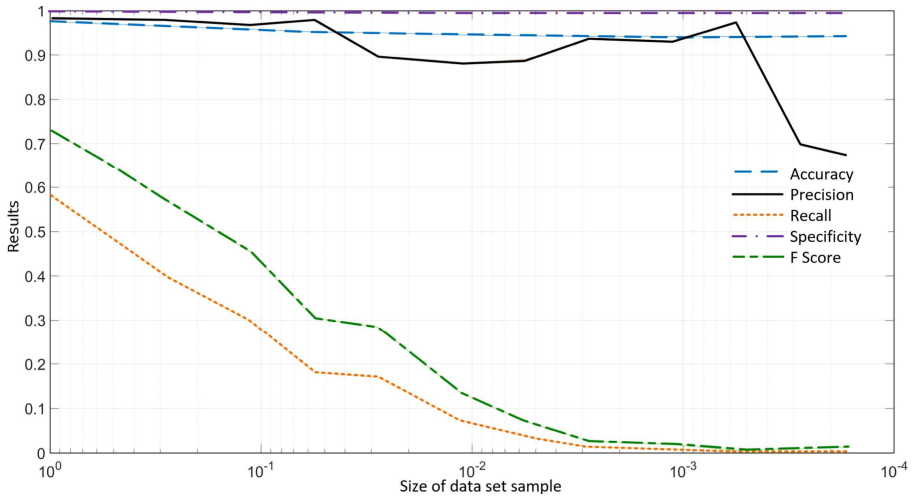

Fig. 13. Effect of changing the number of records on SVM model; ST4000DM000 dataset.

Figure 13 contains the results for the SVM algorithm and the ST4000DM000 dataset. Due to the small number of class 1 records (indicating a malfunction), each time the set is reduced, the failure prediction ability is greatly affected. The results shown in Fig. 14 are very much similar to those obtained for the DT algorithm. Overall, the SVM algorithm also performed well in prediction for the reduced number of records of the CICIDS2017 set.

Figure 15 shows the effect of data sampling on the performance of the k-NN model and the ST4000DM000 dataset. A rapid decrease in the values of precision, recall and F-Score were observed. Compared to the DT algorithm, a much faster decrease in values can be seen. In addition, the accuracy and specificity measures record a slow decrease towards results that are meaningless (form the perspective of these measures).

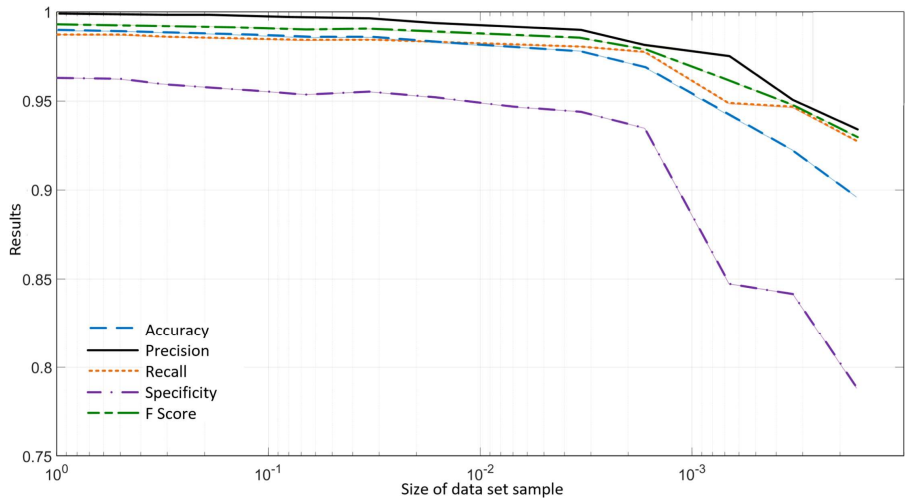

Fig. 14. Effect of changing the number of records on SVM model; CICIDS2017 dataset.

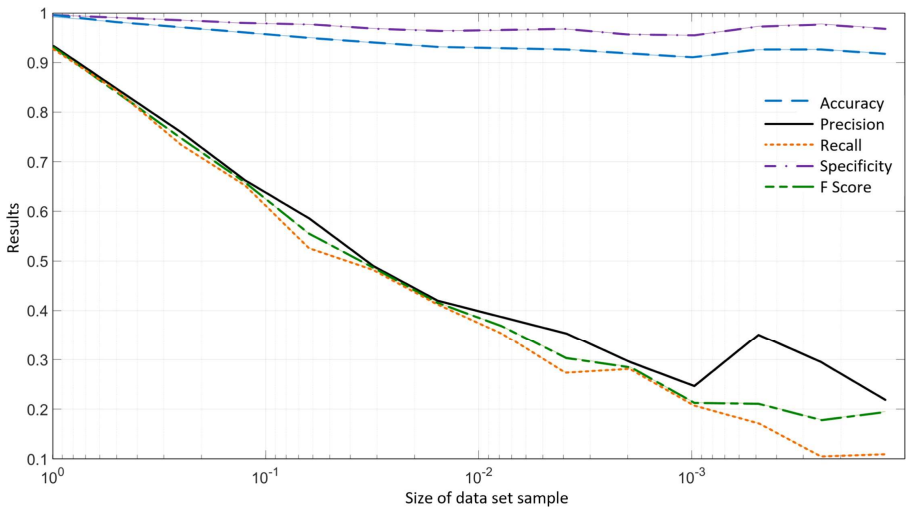

Fig. 15. Effect of changing the number of records on k-NN model; ST4000DM000 dataset.

For the k-NN learning model and the CICIDS2017 dataset (Fig. 16), similar results as for the DT algorithm are observed. Only when the learning set is significantly reduced, there is a greater decrease in the performance measures. The largest performance drop is observed for the specificity measure.

For the NB algorithm, the use of data sampling proved to be problematic due to strong assumptions about the learning datasets, leading to problems with missing variance, for some of the attributes. Due to this error, data sampling was performed to the maximum possible value. Very unsatisfactory results were obtained for the ST4000DM000 set (Fig. 17). The accuracy and specificity remain fairly constant while the model is unable to correctly classify of the possible failures due to the value of the sensitivity measure.

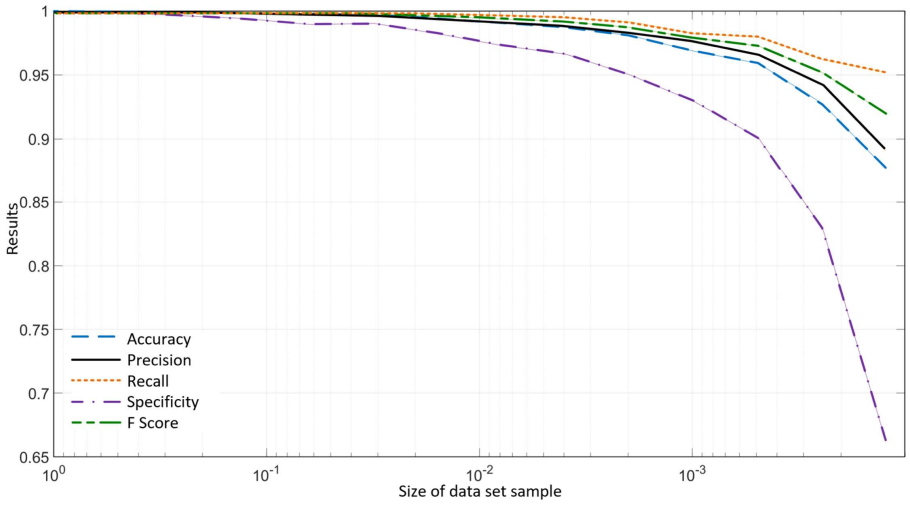

Fig. 16. Effect of changing the number of records on k-NN model; CICIDS2017 dataset

For the CICIDS2017 data, the NB algorithm keeps the performance measure values constant, for the reduced amount of learning data (Fig. 18). The largest increase was recorded for the specificity measure for a learning set size below 10% of the original size when at the same time the other measures recorded a small increase.

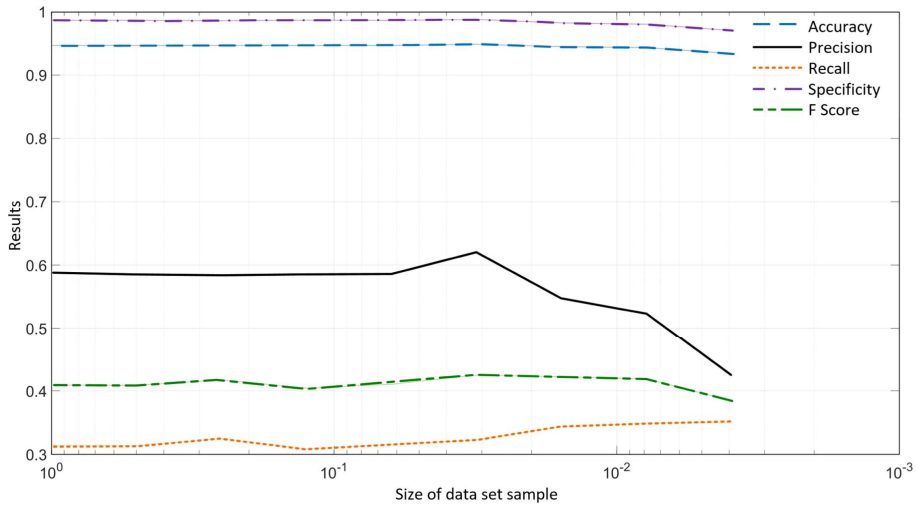

Fig. 17. Effect of changing the number of records on NB model; ST4000DM000 dataset.

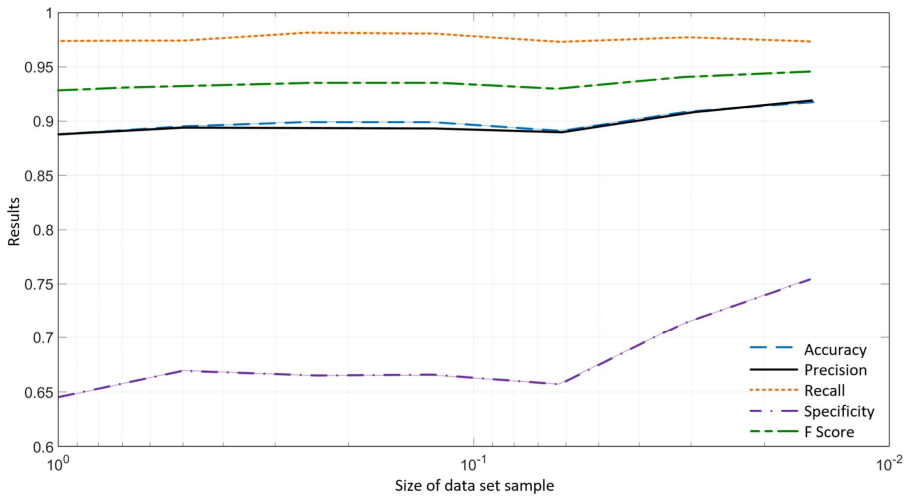

Fig. 18. Effect of changing the number of records on NB model; CICIDS2017 dataset.

The obtained results show that it is possible to reduce the sampling frequency of the learning set data, while maintaining threshold that would be satisfactory for detection and prediction of anomalies, when applied within the first step of the cascade. By using this approach, it is possible not only to reduce the amount of data sent over the networks, but also reduce the learning time. We illustrate this in Fig. 19. Here, we depict the effect of sampling frequency on the learning time of the DT model, for the CICIDS2017 dataset.

It should be noted (again) that the ability to reduce the dataset, while maintaining satisfactory results is strictly dependent on the class of problems. The CICIDS2017 dataset seems to be more amenable to sampling rate reduction when DT, k-NN or SVM algorithms are used. For the ST4000DM00 dataset, the best results are obtained for the k-NN algorithm, but the susceptibility of this dataset to reduce the learning set, while maintaining acceptable failure detection results is low compared to the results obtained for the CICIDS2017 dataset.

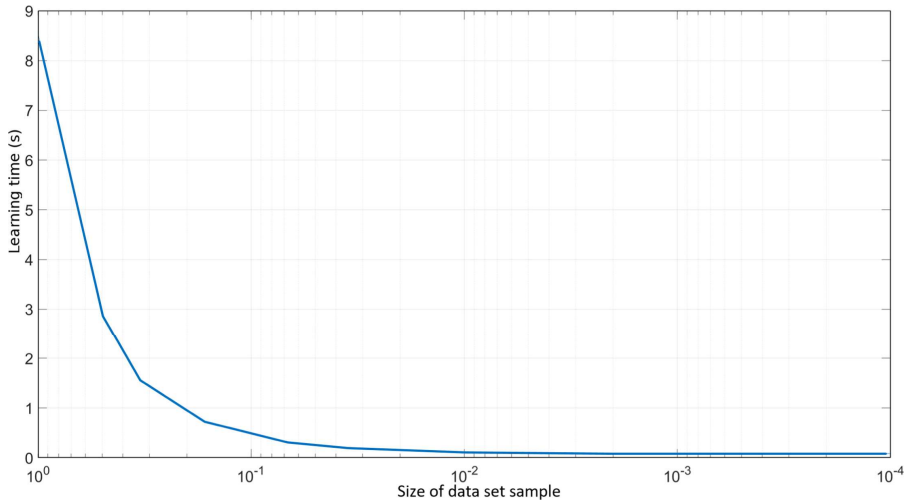

Fig. 19. Effect of changing the number of records on the learning time of the DT model; CICIDS2017 dataset.

5 Concluding Remarks

The results presented in this work represent a contribution to the development of a novel architecture for distributed anomaly detection and prediction in a distributed system. The proposed solution is based on a cascade model, in which the first detection step can be applied, for instance, to non-key systems, which often are not monitored. Classic ML algorithms were applied for anomaly detection. Experimental work was focused on studying effects of reducing (1) number of measured attributes, and (2) data sampling rate, on the efficiency of threat detection by the first level of detectors (RA systems). Obtained results show that it is possible to use limited datasets without significant degradation of anomaly detection performance. However, a significant influence on the results has matching of applied algorithms with the monitored element. Moreover, the specifics of the monitored system are going to determine the minimum acceptable threshold for anomaly detection. Achieved results are rather promising, however, it is necessary to further continue research in this area, using more datasets and more recently proposed ML approaches in order to further experimentally ground the proposed approach and to provide guidelines for its applicability.

In this context, the proposed solution will be implemented and tried in a production system, with large number of disks and auxiliary systems. The work will also focus on developing effective mechanisms for automatic management and reconfiguration of distributed system elements in response to anomalies detected by the proposed approach. Here, preventive measures applicable at each step of the cascade, for different contexts will be formulated.

Acknowledgement. This project is financed by the Minister of Education and Science of the Republic of Poland within the "Regional Initiative of Excellence" program for years 2019–2022. Project number 027/RID/2018/19, amount granted 11 999 900 PLN.

This work has been supported by the joint research project "Agent Technologies in Dynamics Environments" under the agreement on scientific cooperation between University of Novi Sad, University of Craiova, SRI PAS and Warsaw University of Technology, as well as by the joint

research project “Novel methods for development of distributed systems” under the agreement on scientific cooperation between the Polish Academy of Sciences and Romanian Academy for years 2019–2021. Finally, support from the Bulgarian Academy of Sciences and the Polish Academy of Sciences, (Bilateral grant agreement between BAS and PAS) is acknowledged.

References

1. Janus, P., Ganzha, M., Bicki, A., Paprzycki, M.: Applying machine learning to study infrastructure anomalies in a mid-size data center - preliminary considerations. In: Proceedings of the 54th Hawaii International Conference on System Sciences (2021). http://hdl.handle.net/10125/70636. https://doi.org/10.24251/HICSS.2021.025
2. Neu, D.A., Lahann, J., Fettke, P.: A systematic literature review on state-of-the-art deep learning methods for process prediction. Artif. Intell. Rev. (2021). https://doi.org/10.1007/s10462-021-09960-8
3. Williams, A.W., Pertet, S.M., Narasimhan, P.: Tiresias: black-box failure prediction in distributed systems. In: Proceedings of the 2007 IEEE International Parallel and Distributed Processing Symposium, Long Beach, CA, USA, pp. 1–8. IEEE (2007). https://doi.org/10.1109/IPDPS.2007.370345
4. Mariani, L., Pezzè, M., Riganelli, O., Xin, R.: Predicting failures in multi-tier distributed systems. J. Syst. Softw. **161**, 110464 (2020). https://doi.org/10.1016/j.jss.2019.110464
5. Chen, X., Lu, C., Pattabiraman, K.: Failure prediction of jobs in compute clouds: a google cluster case study. In: Proceedings of the 2014 IEEE International Symposium on Software Reliability Engineering Workshops, Naples, Italy, pp. 341–346. IEEE (2014). https://doi.org/10.1109/ISSREW.2014.105
6. Zhao, J., Ding, Y., Zhai, Y., Jiang, Y., Zhai, Y., Hu, M.: Explore unlabeled big data learning to online failure prediction in safety-aware cloud environment. J. Parallel Distrib. Comput. **153**, 53–63 (2021). https://doi.org/10.1016/j.jpdc.2021.02.025
7. https://medium.com/apprentice-journal/pca-application-in-machine-learning-4827c07a61db. Accessed 18 Nov 2021
8. Chigurupati, A., Thibaux, R., Lassar, N.: Predicting hardware failure using machine learning. In: Proceedings of the 2016 Annual Reliability and Maintainability Symposium (RAMS), Tucson, AZ, USA, pp. 1–6 (2016). https://doi.org/10.1109/RAMS.2016.7448033
9. Suchatpong, T., Bhumkittipich, K.: Hard Disk Drive failure mode prediction based on industrial standard using decision tree learning. In: Proceedings of the 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Nakhon Ratchasima, Thailand, pp. 1–4. IEEE (2014). https://doi.org/10.1109/ECTICon.2014.6839839
10. Strom, B.D., Lee, S.C., Tyndall, G.W., Khurshudov, A.: Hard disk drive reliability modeling and failure prediction. In: Proceedings of the Asia-Pacific Magnetic Recording Conference 2006, Singapore, pp. 1–2. IEEE (2006). https://doi.org/10.1109/APMRC.2006.365900
11. Hu, L., Han, L., Xu, Z., Jiang, T., Qi, H.: A disk failure prediction method based on LSTM network due to its individual specificity. Proc. Comput. Sci. **176**, 791–799 (2020). https://doi.org/10.1016/j.procs.2020.09.074
12. Li, Q., Li, H., Zhang, K.: Prediction of HDD failures by ensemble learning. In: Proceedings of the 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, pp. 237–240. IEEE (2019). https://doi.org/10.1109/ICSESS47205.2019.9040739

13. Zhang, S., Wang, Y., Liu, M., Bao, Z.: Data-based line trip fault prediction in power systems using LSTM networks and SVM. IEEE Access **6**, 7675–7686 (2018). https://doi.org/10.1109/ACCESS.2017.2785763
14. Omran, S., El Houby, E.M.F.: Prediction of electrical power disturbances using machine learning techniques. J. Amb. Intel. Hum. Comput. **11**(7), 2987–3003 (2019). https://doi.org/10.1007/s12652-019-01440-w
15. Mehlo, N.A., Pretorius, J.H.C., Rhyn, P.V.: Reliability assessment of medium voltage underground cable network using a failure prediction method. In: Proceedings of the 2019 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Macao, China, pp. 1–5. IEEE (2019). https://doi.org/10.1109/APPEEC45492.2019.8994720
16. Sachan, S., Zhou, C., Bevan G., Alkali, B.: Failure prediction of power cables using failure history and operational condition. In: Procedings of the 2015 IEEE 11th International Conference on the Properties and Applications of Dielectric Materials (ICPADM), Sydney, NSW, Australia, pp. 380–383. IEEE (2015). https://doi.org/10.1109/ICPADM.2015.7295288
17. Kwon, J.-H., Kim, E.-J.: Failure prediction model using iterative feature selection for industrial internet of things. Symmetry **12**, 454 (2020). https://doi.org/10.3390/sym12030454
18. Fernandes, S., Antunes, M., Santiago, A.R., Barraca, J.P., Gomes, D., Aguiar, R.L.: Forecasting appliances failures: a machine-learning approach to predictive maintenance. Information **11**, 208 (2020). https://doi.org/10.3390/info11040208
19. Cai, Z., Sun, S., Si, S., Wang, N.: Research of failure prediction Bayesian network model. In: Proceedings of the 2009 16th International Conference on Industrial Engineering and Engineering Management, Beijing, China, pp. 2021–2025. IEEE (2009). https://doi.org/10.1016/10.1109/ICIEEM.2009.5344265
20. Bai, C.G., Hu, Q.P., Xie, M., Ng, S.H.: Software failure prediction based on a Markov Bayesian network model. J. Syst. Softw. **74**(3), 275–282 (2005). https://doi.org/10.1016/j.jss.2004.02.028
21. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network Traffic Anomaly Detection and Prevention: Concepts, Techniques, and Tools. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-65188-0. ISBN 978-3-319-87968-0
22. Bolanowski, M., Twaróg, B., Mlicki, R.: Anomalies detection in computer networks with the use of SDN. Meas. Autom. Monit. **9**(61), 443–445 (2015)
23. Bolanowski, M., Paszkiewicz, A.: The use of statistical signatures to detect anomalies in computer network. In: Gołębiowski, L., Mazur, D. (eds.) Analysis and Simulation of Electrical and Computer Systems. LNEE, vol. 324, pp. 251–260. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-11248-0_19
24. Zhong, J., Guo, W., Wang, Z.: Study on network failure prediction based on alarm logs. In: Proceedings of the 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), Muscat, Oman, pp. 1–7. IEEE (2016). https://doi.org/10.1109/ICBDSC.2016.7460337
25. Ji, W., Duan, S., Chen, R., Wang, S., Ling, Q.: A CNN-based network failure prediction method with logs. In: Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, pp. 4087–4090. IEEE (2018). https://doi.org/10.1109/CCDC.2018.8407833
26. Bolanowski, M., Paszkiewicz, A., Rumak, B.: Coarse traffic classification for high-bandwidth connections in a computer network using deep learning techniques. In: Barolli, L., Yim, K., Enokido, T. (eds.) CISIS 2021. LNNS, vol. 278, pp. 131–141. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79725-6_13
27. Bolanowski, M., Paszkiewicz, A., Kwater, T., Kwiatkowski, B.: The multilayer complex network design with use of the arbiter. Monographs in Applied Informatics, Computing in Science and Technology, Wydawnictwo Uniwersytetu Rzeszowskiego, Rzeszów, pp. 116–127 (2015). ISBN 978-83-7996-140-5

28. https://pypi.org/project/netmiko. Accessed 8 Nov 2021
29. https://www.paramiko.org. Accessed 8 Nov 2021
30. https://www.mathworks.com/help/stats/getting-started-12.html. Accessed 8 Nov 2021
31. https://www.backblaze.com/b2/hard-drive-test-data.html. Accessed 8 Nov 2021
32. Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP, Funchal, Madeira, Portugal, pp. 108–116. SciTePress (2018). https://doi.org/10.5220/0006639801080116