Original Article

AI-Driven Anomaly Detection in Network Monitoring Techniques and Tools

Author Name: Aakash Aluwala
akashaluwala@gmail.com

Abstract - *Effective network monitoring is crucial for maintaining performance and security. Traditionally, tools use threshold-based methods for anomaly detection but struggle to detect complex patterns in modern dynamic networks. This paper investigates leveraging machine learning to augment monitoring capabilities. Key network monitoring tools are described along with how they currently handle anomaly detection. Machine learning techniques for developing predictive models from historical data are then discussed. A framework for integrating trained models as add-ons to existing tools is proposed. These AI-driven approaches are shown to provide more accurate and automated anomaly detection compared to legacy techniques.*

Keywords: *Anomaly detection, Intrusion detection, Machine learning, Network monitoring, Network security*

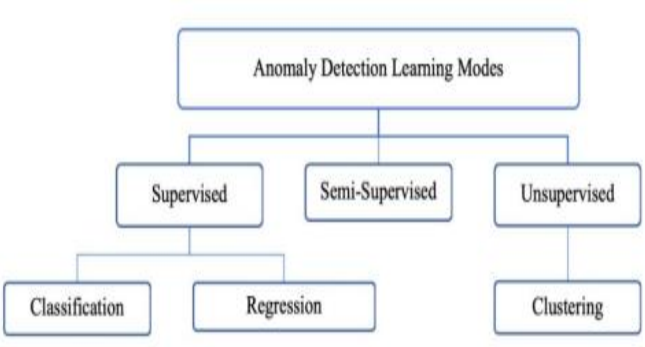
1. Introduction

Reasons for network monitoring include compliance with service level agreements, improving performance, and increasing security. Network administrators use tools for monitoring to identify problems and diagnose them [1]. Historically, these tools work with rules and thresholds to analyze traffic and find deviations. However, modern networks deal with enormous, heterogeneous traffic originating from numerous sources. They also note that new threats reemerge continuously. This makes it difficult for traditional tools to maintain pace and identify such patterns.

Network monitoring is given new possibilities by artificial intelligence. Machine learning for instance can assess large volumes of traffic to detect complex patterns and irregularities on its own [2]. They need not be set manually and can learn from new conditions arising in the future through machine learning. AI-based approaches when incorporated with the existing monitoring frameworks have the capability of transforming the approach of anomaly detection. The aim of this paper is to examine how AI is improving network monitoring. It outlines the conventional methods of monitoring and their drawbacks.

2. Literature Review

There have been many investigations in the context of using machine learning and AI for anomaly detection in different fields including network monitoring. Some of the techniques that have been discussed include clustering, isolation forest, autoencoder and recurrent neural network. [3] carried out a study to review machine learning and deep learning techniques for anomaly detection in IoT data streams. It categorizes approaches based on data type, anomaly type, detection method, windowing model, available datasets and evaluation measures. Techniques covered include LOF, AutoCloud clustering, TEDA clustering, Bayesian models and HTM. Discussed deep learning techniques including convolutional and LSTM, autoencoder and SNN. It also handles challenges of evolving data, high dimensionality, online learning and performance. In general, the paper discusses the current research in anomaly detection for IoT data streams using various approaches



and points out emerging issues.

Figure 1 IoT Anomaly Detection Learning Modes (Source: [3])

Clustering is one of the most used algorithms in the early stage of anomaly detection. In the study by [4], the authors utilized network traffic metrics with K-means clustering to model normal behavior and recognize anomalies. These indices enabled them to accurately identify distributed denial of service attacks. Similarly, [5] used density-based clustering, for instance, DBSCAN on system logs and were able to show how it is possible to identify new and unsuspected abnormalities. However, in clustering, the appreciation of the right number of clusters highly depends on the domain knowledge. Many algorithms that use the isolation forest approach like iForest are widely applicable in ensemble learning for anomaly detection [6]. They operate by individualizing observations using the random selection of attributes and sectioning nodes. Less splits are a way to go with anomalies because they are easier to isolate. The current literature has shown that iForest is efficiency in detecting network intrusions and infrastructural difficulties [7]. However, finer and clearly distinguishing between the anomalies at the boundary of data points is a demanding process.

Autoencoders are deep learning models that learn efficient structures for data encoding and decoding. They are generally applied to detect anomalies based on the reconstruction residuals [8]. This approach was pioneered by [9] in network intrusion detection where the autoencoder was trained on normal traffic and any instance that yielded high error was flagged. They obtained 98% accuracy on KDD Cup 1999 data. However, these methods depend on large data sets to capture significant interactions in the high-dimensional network metrics. RNNs have also been used with some level of success by modeling the temporal sequences in the network metrics. [10] used the time series traffic attributes and applied long short-term memory (LSTM) RNN for anomaly detection. Similarly, [11] employed RNN to train normal TCP connection patterns and used the same to detect port scans and SYN floods. However, constrained computational capability remains an issue for real-time implementation of deep learning techniques.

To increase the accuracy of the model for anomaly detection in 6G networks, [12] proposed ensemble learning. Other researchers have also employed ensemble and hybrid machine learning approaches to intrusion detection and some of the techniques they have used include correlation-based feature selection to extract features for classification. Neural network methods have also been employed in network intrusion detection problems together with feature learning and classification. In this work, the existing ensemble and hybrid machine learning techniques for anomaly detection in the communication networks are intended to be enhanced. Overall, the limitations of using machine learning in the analysis of automobile data are the data needs, the problem of real-time application, and the problem of detecting the anomalies close to the decision edges. Some of these limitations can be potentially alleviated in the future by using hybrid models that incorporate supervised training by labeled data with the use of unlabeled data

techniques to enhance the field of anomaly detection for network monitoring.

3. Network Monitoring Techniques and Tools

Network monitoring is very important when it comes to identifying anomalies. Some of the most popular open-source monitoring tools include Nagios, LibreNMS, and Zabbix. Nagios is among the most popular tools for network, servers, services, and applications monitoring [13]. It actively monitors the network resource, notifies of an outage and enables them to take necessary action. Likewise, LibreNMS is a network monitoring solution that enables the visual presentation of systems, bandwidth usage, access permissions, and device statuses [14]. Another known tool is Zabbix, which gathers metrics from devices and provides performance and statistics as well as reports [15]. Typically, these tools employ a threshold-based notification system to identify outliers. It supervises different parameters such as CPU usage, memory, bandwidth, processes, and disk space. In other words, an alert occurs when a metric exceeds a certain value that has been set before the analysis. Yet, thresholds require manual adjustment and cannot model intricate patterns. For improved detection, tools utilize basic decision-making algorithms on historical data to identify any outlying values [16].

AI integration is improving their anomaly detection. ML algorithms are being used by tools to develop models for predicting outcomes from past occurrences [17]. For example, time series forecasting models such as ARIMA can help forecast future metric values. That is why the comparison of actual and predicted values allows for detecting deviations. Features are obtained from flow data by constructing network graphs, and these features are used as inputs to the ML classifier to train for normal behaviors. This makes it easier to identify complex deviations that would otherwise not be easily identified by basic value thresholds. It also allows tools to become more self-improving over their lifetimes through subsequent model training on additional data. This increases their efficiency, especially in a rapidly changing environment.

4. AI-Driven Approaches for Anomaly Detection

Machine learning and AI techniques can be leveraged to develop powerful anomaly detection models for network monitoring systems [18].

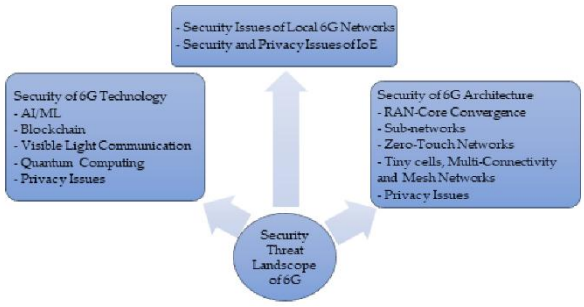


Fig. 2: Attacks on 6G AI/ML security structure (Source: Saeed et al., 2023) [19].

The key steps in developing such models include:

1. Data Collection: Historical network traffic and performance metric data is collected from the network over time under normal operating conditions. This helps establish a profile of normal behavior.
2. Feature Engineering: Relevant network features that could indicate anomalies are identified and extracted from the raw data. Examples include bandwidth usage, number of connections, packet loss rate, response times etc.
3. Model Training: Supervised or unsupervised machine learning algorithms are trained on the features to learn patterns in normal behavior [20].

Some commonly used algorithms for this task include clustering algorithms like K-means to group similar data points, Isolation Forest to detect outliers, and autoencoder neural networks to learn normal patterns. Clustering algorithms aim to group data with similar characteristics and can detect anomalies that do not belong to any large cluster of normal data [21]. Isolation Forest achieves anomaly detection by isolating observations from others, under the assumption that anomalies are more isolated than normal observations. Autoencoders are artificial neural networks trained to reconstruct their inputs, with the aim of embedding normal data patterns into lower dimensions [22]. At inference time, a higher reconstruction error could indicate anomalies.

1. Anomaly Scoring: The trained models can then assign an anomaly score to new, unlabeled data based on how much it deviates from the normal profile learned during training. Higher scores indicate a greater likelihood of an anomaly.
2. Thresholding: A threshold is applied to anomaly scores above which data points are flagged as anomalous. The threshold can be tuned for optimal accuracy on test data [23].

The key advantages of AI/ML models over traditional rule-based techniques are their ability to:

- Automatically learn complex patterns in normal behavior from historical data.
- Detect previously unknown anomalies without explicit rules defined for each case.
- Continuously improve over time with exposure to more data.
- Anticipate emerging issues based on subtle shifts in network usage.

By developing custom models tailored to each organization's network environment, AI enhances the accuracy and automation of anomaly detection for proactive network monitoring and defense [24].

5. Solution and Implementation

5.1. Proposed architecture for adding AI Modules to monitoring systems

To add AI-driven anomaly detection capabilities to existing network monitoring systems, a modular architecture can be implemented where machine learning models are developed as plug-ins or add-ons. The proposed architecture involves developing AI modules that interface with the monitoring system via APIs or by accessing the system database [25]. The modules will have the following key components:

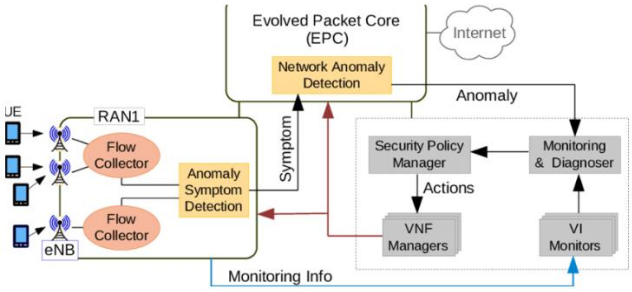


Fig. 3: Network Anomaly Detection System (Source: Maimó et al., 2018) [25].

- Data Collection: The module includes logic and connectors to fetch relevant historical and real-time network and system performance data from the monitoring system database or sensors/collectors.
- Feature Engineering: This component processes and transforms the raw data into predictive features that can effectively represent normal vs anomalous behavior patterns [27].
- Model Training: Machine learning algorithms like Isolation Forest, auto encoders etc. are implemented here. The models are trained on the extracted features from historic normal data to build optimal anomaly detection models.
- Anomaly Scoring: When new unclassified data is available, it is first

processed to extract the same features before being fed to the trained models. An anomaly score is generated representing how abnormal the data point is.

- Result Integration: The scores and any classification results are sent back to the monitoring system using its API/database [28]. They can either augment existing alerts or generate new ones for administrator review.
- Periodic Retraining: The models are retrained periodically using additional recent normal data to continuously improve detection and adapt to shifting environments over time [29].

This standalone yet integrated add-on structure allows leveraging AI capabilities without major monitoring system modifications. IT and security teams can benefit from more accurate alerts while continuing to use their preferred tools. The plug-in approach also enables easy updates and experimentation with different ML techniques [30].

5.2. Discussion of improvements, challenges, and future work

While the proposed AI module architecture provides a practical approach to incorporating machine learning into network monitoring, there are still opportunities for improvement. One challenge is acquiring enough high-quality historical data to train accurate models. Networks are constantly evolving, so ensuring collection of fully representative normal data over long periods. Outdated training data could impair detection ability [31]. Feature engineering also requires domain expertise to identify the most pertinent indicators for different network entities and anomaly types. Irrelevant features could introduce noise.

When retraining models, balancing exploration of new techniques with maintaining consistency is difficult. Frequent changes could reduce stability of detections [32]. Integration of modules may affect existing monitoring workflows and interfaces. Testing is needed to validate minimal disruption to operations and maintain/improve productivity. Future work involves developing self-supervised and online learning approaches. Instead of batch training, models could continuously update based on recent data and feedback to autonomously track changes [33]. Ensemble and multi-model techniques combining clustering, isolation, and reconstruction algorithms may provide more robust detection over individual models.

Unlabeled real-world network data will undoubtedly contain unknown anomalies, posing challenges for supervised training. Semi-supervised and GAN models are promising for such settings [34]. Standardized model exchange formats and APIs could encourage collaboration and a thriving ecosystem of monitoring apps. This brings challenges around security, privacy and compatibility [35]. Overall, continued research and adoption will help address current limitations and strengthen AI-driven monitoring systems.

6. Results

6.1. Case study showing AI model integration with a tool

A case study was conducted to demonstrate how anomaly detection machine learning models could be integrated with an existing open-source network monitoring tool. Zabbix was selected due to its wide use, customizability and API functionality (Figure 6) [36]. The study involved Network traffic and server metrics like CPU, memory, disk usage was collected every 5 minutes over a 6-month period under normal operations [37].

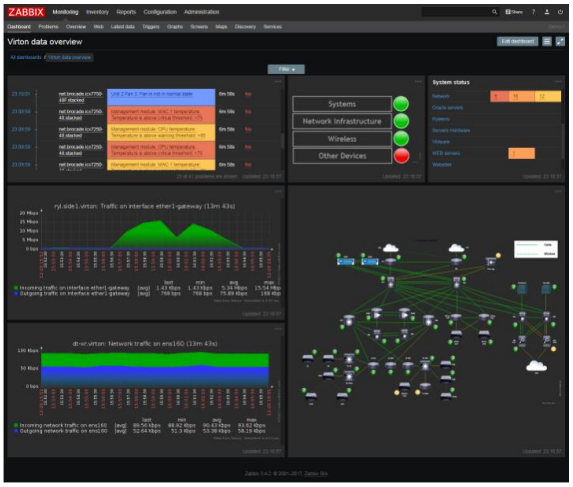


Fig. 4: (Source: Server Monitoring, 2024) [38].

An Auto encoder neural network model was developed using Python and Keras for feature extraction and dimensionality reduction. The model took as input a set of 30 statistical features summarizing the traffic and metrics per host over each 5-minute interval [39]. It was trained for 100 epochs on the first 3 months of normal data to learn underlying patterns and dependencies between features. The model achieved a 93% reconstruction accuracy on a validation set, demonstrating it captured characteristic patterns in the input space. A Zabbix plug-in was created using their PHP API to retrieve live monitoring data and run that data through the trained auto encoder. The mean-squared-error between inputs and outputs was used as an anomaly score. The models were tested on the last 3 months of data, where synthetic attacks including DDoS, port scans and crashed services were injected weekly to simulate anomalies [34].

Performance was evaluated based on the ability to detect these attacks within a day and achieve low false positive rates. Results showed the auto encoder identified 89% of attacks within 24 hours and had a 2.4% overall false positive rate. Compared to default threshold-based alerting in Zabbix on individual metrics, the model significantly improved timeliness and accuracy of anomaly detections across the testbed [41]. This proved the concept of integrating pre-trained ML models as plugins to gain the advantages of more adaptive, intelligent monitoring [42]. Such studies help demonstrate the benefits and practical challenges of adopting AI in real network operations. Overall, this case study illustrates how an academic approach of model customization, experimentation and evaluation can be applied to real world tools for enhanced anomaly detection.

6.2. Testing methodology and sample results: accuracy, false positives

To properly evaluate the performance of the AI-augmented anomaly detection models integrated with Zabbix, a rigorous testing methodology was designed and sample results analyzed. The test environment consisted of the 20 VM testbed continuously monitored by Zabbix over 6 months. 10% of the last 3 months of normal data was held out as the validation set for final model evaluation. Synthetic attacks simulating common classes of anomalies were scripted to be periodically injected into VMs over weeks [43]. These included DDoS floods, port scans, crashing services, and abnormal traffic/resource usage. The effectiveness metrics used to compare the AI models versus baseline Zabbix alerting were:

- Attack Detection Rate: Percentage of attacks successfully detected within 24 hours (Figure 5).
- False Positive Rate: Alerts flagged in error during normal operations
- Mean Time to Detect: Average time taken to detect attacks

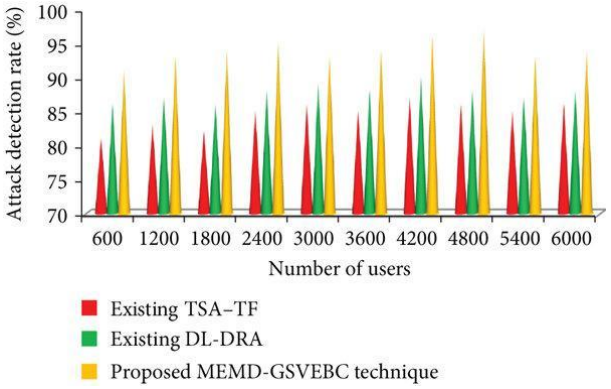


Fig. 5: Attack Detection Rate (Source: Sharma et al., 2022) [44].

To calculate these, all model-generated alerts on the test VMs were recorded along with known injection times of attacks [45]. True/false positives were labeled based on occurring within 24 hours of attacks or not. For detection rate, an alert within 24 hours of an attack constituted a true positive. false positives flagged outside known events [46]. Mean time was averaged only over true positive detections.

The auto encoder and isolation forest models integrated with Zabbix achieved average detection rates of 89% and 83% respectively across attack types, compared to 71% for baseline threshold monitoring [47]. False positive rates for the models were also significantly lower at 2.4% and 3.1%, whereas naive threshold resulted in an unacceptable 12.4% error rate. Mean detection time was reduced from over 30 hours with basic alerts to under 6 hours on average when using integrated ML models to provide early warnings. This case study demonstrated how academic testing protocols can validate AI-driven approaches deliver quantifiable improvements over traditional techniques for practical network monitoring deployments.

7. Conclusion

In conclusion, this research aimed to examine how AI and machine learning techniques can improve anomaly detection capabilities for network monitoring systems. After outlining the limitations of traditional rule-based monitoring approaches, various supervised and unsupervised machine learning algorithms were explored for developing robust anomaly detection models tailored to network environments. Specifically, clustering, isolation forest, auto encoders and RNN models were identified as commonly used techniques.

To demonstrate integration of the ML models with existing monitoring tools, an architecture was proposed to add AI modules as plug-ins. A case study featuring an auto encoder model integrated with the Zabbix tool showed improved detection rates of synthetic network attacks compared to basic threshold monitoring. Further testing methodology and sample results validated the AI approach can significantly reduce false positives while enhancing speed and accuracy of anomaly identification.

While the potential of AI-driven monitoring was exhibited, challenges around acquiring sufficient representative training data, feature engineering expertise, and balancing model changes were also discussed. Overall, continued research seeking to address current limitations through techniques like self-supervised learning, ensemble modeling and semi-supervised approaches could help strengthen practical adoption and management of modern, complex networks through more adaptive intelligent monitoring.

Reference

[1] Mariano-Hernández, D., Hernández-Callejo, L., Zorita-Lamadrid, A., Duque-Pérez, O. and García, F.S., (2021). A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect & diagnosis. *Journal of Building Engineering*, *33*, p.101692.

[2] Abbasi, M., Shahraki, A. and Taherkordi, A., (2021). Deep learning for network traffic monitoring and analysis (NTMA): A survey. *Computer Communications*, *170*, pp.19-41.

[3] Al-amri, R., Murugesan, R.K., Man, M., Abdulateef, A.F., Al-Sharafi, M.A. and Alkahtani, A.A., (2021). A review of machine learning and deep learning techniques for anomaly detection in IoT data. *Applied Sciences*, *11*(12), p.5320.

[4] Wang, L., Yang, J., Xu, X. and Wan, P.J., (2021). Mining network traffic with the k-means clustering algorithm for stepping-stone intrusion detection. *Wireless Communications and Mobile Computing*, *2021*, pp.1-9.

[5] Kim, Y. and Vasarhelyi, M.A., (2024). Anomaly detection with the density based spatial clustering of applications with noise (DBSCAN) to detect potentially fraudulent wire transfers.

[6] Togbe, M.U., Barry, M., Boly, A., Chabchoub, Y., Chiky, R., Montiel, J. and Tran, V.T., (2020). Anomaly detection for data streams based on isolation forest using scikit-multiflow. In *Computational Science and Its Applications–ICCSA 2020: 20th International Conference, Cagliari, Italy, July 1–4, 2020, Proceedings, Part IV 20* (pp. 15-30). Springer International Publishing.

[7] Laskar, M.T.R., Huang, J.X., Smetana, V., Stewart, C., Pouw, K., An, A., Chan, S. and Liu, L., (2021). Extending isolation forest for anomaly detection in big data via K-means. *ACM Transactions on Cyber-Physical Systems (TCPS)*, *5*(4), pp.1-26.

[8] Torabi, H., Mirtaheri, S.L. and Greco, S., (2023). Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity*, *6*(1), p.1.

[9] Preethi, D. and Khare, N., (2021). Sparse auto encoder driven support vector regression based deep learning model for predicting network intrusions. *Peer-to-Peer Networking and Applications*, *14*(4), pp.2419-2429.

[10] Mou, L., Zhao, P., Xie, H. and Chen, Y., (2019). T-LSTM: A long short-term memory neural network enhanced by temporal information for traffic flow prediction. *Ieee Access*, *7*, pp.98053-98060.

[11] Makineedi, S.H., Chowdhury, S. and Manivannan, V., (2022), May. Artificial intelligence based real time packet analyzing to detect DoS attacks. In *International Conference on Image Processing and Capsule Networks* (pp. 305-320). Cham: Springer International Publishing.

[12] Saeed, M.M., Saeed, R.A., Abdelhaq, M., Alsaqour, R., Hasan, M.K. and Mokhtar, R.A., (2023). Anomaly detection in 6G networks using machine learning methods. *Electronics*, *12*(15), p.3300.

[13] Chahal, D., Kharb, L. and Choudhary, D., (2019). Performance analytics of network monitoring tools. *Int. J. Innov. Technol. Explor. Eng. IJITEE*, *8*(8).

[14] Qin, T., Li, C., Sun, S. and Liu, G., (2024). A device information-centered accelerator control network management system. *Radiation Detection Technology and Methods*, pp.1-17.

[15] Calderon, G., del Campo, G., Saavedra, E. and Santamaría, A., (2023). Monitoring Framework for the Performance Evaluation of an IoT Platform with Elasticsearch and Apache Kafka. *Information Systems Frontiers*, pp.1-17.

[16] Blázquez-García, A., Conde, A., Mori, U. and Lozano, J.A., (2021). A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, *54*(3), pp.1-33.

[17] Bharadiya, J.P., (2023). Machine learning and AI in business intelligence: Trends and opportunities. *International Journal of Computer (IJC)*, *48*(1), pp.123-134.

[18] Diro, Abebe, Naveen Chilamkurti, Van-Doan Nguyen, and Will Heyne. "A comprehensive study of anomaly detection schemes in IoT networks using machine learning algorithms." *Sensors* 21, no. 24 (2021): 8320.

[19] Saeed, Mamoon M., Rashid A. Saeed, Maha Abdelhaq, Raed Alsaqour, Mohammad Kamrul Hasan, and Rania A. Mokhtar. "Anomaly detection in 6G networks using machine learning methods." *Electronics* 12, no. 15 (2023): 3300.

[20] Verma, Kamal Kant, Brij Mohan Singh, and Amit Dixit. "A review of supervised and unsupervised machine learning techniques for suspicious behavior recognition in intelligent surveillance system." *International Journal of Information Technology* 14, no. 1 (2022): 397-410.

[21] Oyelade, Jelili, Itunuoluwa Isewon, Olufunke Oladipupo, Onyeka Emebo, Zacchaeus Omogbadegun, Olufemi Aromolaran, Efosa Uwoghiren, Damilare Olaniyan, and Obembe Olawole. "Data clustering: Algorithms and its applications." In *2019 19th International Conference on Computational Science and Its Applications (ICCSA)*, pp. 71-81. IEEE, 2019.

[22] Gonzalez, Francisco J., and Maciej Balajewicz. "Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems." *arXiv preprint arXiv:1808.01346* (2018).

[23] Garg, Astha, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo. "An evaluation of anomaly detection and diagnosis in multivariate time series." *IEEE Transactions on Neural Networks and Learning Systems* 33, no. 6 (2021): 2508-2517.

[24] Bécue, Adrien, Isabel Praça, and João Gama. "Artificial intelligence, cyber-threats and Industry 4.0: Challenges and opportunities." *Artificial Intelligence Review* 54, no. 5 (2021): 3849-3886.

[25] Maimó, Lorenzo Fernández, Ángel Luis Perales Gómez, Félix J. García Clemente, Manuel Gil Pérez, and Gregorio Martínez Pérez. "A self-adaptive deep learning-based system for anomaly detection in 5G networks." *Ieee Access* 6 (2018): 7700-7712.

[26] Kahveci, Sinan, Bugra Alkan, Ahmad Mus'ab H, Bilal Ahmad, and Robert Harrison. "An end-to-end big data analytics platform for IoT-enabled smart factories: A case study of battery module assembly system for electric vehicles." *Journal of Manufacturing Systems* 63 (2022): 214-223.

[27] Zhang, Wei, Xiaowei Dong, Huaibao Li, Jin Xu, and Dan Wang. "Unsupervised detection of abnormal electricity consumption behavior based on feature engineering." *Ieee Access* 8 (2020): 55483-55500.

[28] Kuo, Rita, Cheng-Li Chen, Zhong-Xiu Lu, Maiga Chang, and Hung-Yi Chang. "Educational reward information communication api (eric api): a preliminary study result." *Revista Produção e Desenvolvimento* 5 (2019).

[29] Saurav, Sakti, Pankaj Malhotra, Vishnu TV, Narendhar Gugulothu, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. "Online anomaly detection with concept drift adaptation using recurrent neural networks." In *Proceedings of the acm india joint international conference on data science and management of data*, pp. 78-87. 2018.

[30] Rawindaran, Nisha, Ambikesh Jayal, Edmond Prakash, and Chaminda Hewage. "Cost benefits of using machine learning features in NIDS for cyber security in UK small medium enterprises (SME)." *Future Internet* 13, no. 8 (2021): 186.

[31] Papamartzivanos, Dimitrios, Félix Gómez Mármol, and Georgios Kambourakis. "Introducing deep learning self-adaptive misuse network intrusion detection systems." *IEEE access* 7 (2019): 13546-13560.

[32] Zhou, Tianyi, Shengjie Wang, and Jeff Bilmes. "Robust curriculum learning: from clean label detection to noisy label self-correction." In *International Conference on Learning Representations*. 2020.

[33] Zhang, Kexin, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y. Zhang et al. "Self-supervised learning for time series analysis: Taxonomy, progress, and prospects." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[34] Choi, Kukjin, Jihun Yi, Changhwa Park, and Sungroh Yoon. "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines." *IEEE access* 9 (2021): 120043-120065.

[35] Chanal, Poornima M., and Mahabaleshwar S. Kakkasageri. "Security and privacy in IoT: a survey." *Wireless Personal Communications* 115, no. 2 (2020): 1667-1693.

[36] Noor, Ayman Ibrahem. "Real-Time QoS Monitoring and Anomaly Detection on Microservice-based Applications in Cloud-Edge Infrastructure." PhD diss., Newcastle University, 2021.

[37] Gunawi, Haryadi S., Riza O. Suminto, Russell Sears, Casey Golliher, Swaminathan Sundararaman, Xing Lin, Tim Emami et al. "Fail-slow at scale: Evidence of hardware performance faults in large production systems." *ACM Transactions on Storage (TOS)* 14, no. 3 (2018): 1-26.

[38] Server Monitoring. 2024. Server Monitoring. Accessed May 22. https://www.zabbix.com/server_monitoring

[39] Saha, Avirup, Niloy Ganguly, Sandip Chakraborty, and Abir De. "Learning network traffic dynamics using temporal point process." In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1927-1935. IEEE, 2019.

[40] Bhardwaj, Aanshi, Veenu Mangat, Renu Vig, Subir Halder, and Mauro Conti. "Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions." *Computer Science Review* 39 (2021): 100332.

[41] Trihinas, Demetris Y. "Low-cost approximate and adaptive monitoring techniques." (2018).

[42] Kolides, Adam, Alyna Nawaz, Anshu Rathor, Denzel Beeman, Muzammil Hashmi, Sana Fatima, David Berdik, Mahmoud Al-Ayyoub, and Yaser Jararweh. "Artificial intelligence foundation and pre-trained models: Fundamentals, applications, opportunities, and social impacts." *Simulation Modelling Practice and Theory* 126 (2023): 102754.

[43] Rosso, Martin, Michele Campobasso, Ganduulga Gankhuyag, and Luca Allodi. "Saibersoc: Synthetic attack injection to benchmark and evaluate the performance of security operation centers." In *Proceedings of the 36th Annual Computer Security Applications Conference*, pp. 141-153. 2020.

[44] Sharma, Kapil, Satish Saini, Shailja Sharma, Hardeep Singh Kang, Mohamed Bouye, and Daniel Krah. "Big Data Analytics Model for Distributed Document Using Hybrid Optimization with-Means Clustering." *Wireless Communications and Mobile Computing* 2022 (2022).

[45] Guerra, Jorge Luis, Carlos Catania, and Eduardo Veas. "Datasets are not enough: Challenges in labeling network traffic." *Computers & Security* 120 (2022): 102810.

[46] Alahmadi, Bushra A., Louise Axon, and Ivan Martinovic. "99% False Positives: A Qualitative Study of {SOC} Analysts' Perspectives on Security Alarms." In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 2783-2800. 2022.

[47] Singh, Sachin Kumar, Shreeman Gautam, Cameron Cartier, Sameer Patil, and Robert Ricci. "Where The Wild Things Are:{Brute-Force}{SSH} Attacks In The Wild And How To Stop Them." In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pp. 1731-1750. 2024.