

Detecting and Analyzing Agent Communication Anomalies in Distributed Energy System Control

Emilie Frost^{1,2}^a, Julia Catharina Heiken^{1,2}^b, Martin Tröschel²^c and Astrid Nieße^{1,2}^d

¹Digitalized Energy Systems, Carl von Ossietzky Universität Oldenburg,
Ammerländer Heerstraße 114-118, Oldenburg, Germany

²Distributed Artificial Intelligence, OFFIS e.V., Escherweg 2, Oldenburg, Germany

fi

Keywords: Anomaly Detection, Self-Organizing Systems, Multi-Agent Systems.

Abstract: In Cyber-Physical Energy Systems (CPES), multi-agent systems are expected to perform a variety of tasks. The increase in digital interconnections and distributed structures in CPES leads to more cyber access points, which increases the risk of cyber attacks. The effect of a manipulated or corrupted agent, as caused by cyber attacks, on the communication of an agent system is investigated in this paper. Anomaly detection is an important prerequisite to identify and mitigate malicious behavior and thus protect the critical infrastructure of CPES. Since in distributed systems, some information is only available in a distributed way, this paper introduces a centralized and a distributed architecture for anomaly detection. For this, a dataset is presented from an agent-based energy system control use case, including anomalies in agent behavior.


1 INTRODUCTION


The number of digitalized control systems continues to rise in current power systems (Chen et al., 2012). Due to the increasing importance of intelligent automation and the accompanying growth of communication technologies, more interconnections exist between physical and cyber components, resulting in Cyber-Physical Energy Systems (CPES). Consequently, risks increasingly originate from the cyberspace part of the CPES. Therefore, the strong interconnections in CPES lead to new challenges for the communication needs (Chen et al., 2012). These challenges also refer to agent-based control systems, which may enable self-organization or even self-healing properties of the system, especially in safety-critical applications (Nieße and Tröschel, 2016; Veith et al., 2014). In these systems, attacks or cyber intruders significantly impact the overall system's performance. This paper examines the effect of a single agent with manipulated behavior on the overall system. The agent negotiates with others and is corrupted in its behavior as if an attacker would have taken over.


Controlled self-organization furthermore allows the monitoring of the self-organizing system to detect deviations from normal behavior or errors (Nieße and Tröschel, 2016). The overall concept stems from the field of Organic Computing (Schmeck et al., 2010). It is essential to detect deviations from normal operations to react to them. Therefore, anomaly detection is a very important prerequisite for protecting the critical infrastructure against the aforementioned threats.


However, detecting anomalies in power systems is a challenge due to the complexity of power system monitoring and control systems and the inherently diffuse data of CPES measurements (Ferragut et al., 2013). Nevertheless, system monitoring is essential to detect anomalies. For this reason, it is necessary to analyze CPES states at multiple scales to ensure consideration of individual components as well as network-level dynamics (Ferragut et al., 2013).

In order to address these challenges, this paper analyzes different anomaly detection approaches. Therefore, a centralized and a distributed architecture are implemented, considering an agent-based application within a CPES. Certain information may not be shared in distributed systems for e.g. privacy reasons, which may limit the information for the anomaly detection. For this reason, the effect of information availability on the performance of anomaly detection is additionally analyzed.

^a <https://orcid.org/0000-0003-4791-2333>

^b <https://orcid.org/0009-0002-3946-8949>

^c <https://orcid.org/0000-0002-9882-5144>

^d <https://orcid.org/0000-0003-1881-9172>

Within this context, the contribution of this work is as follows:

1. A dataset is presented from an agent-based energy system control use case, including anomalies in agent behavior. The dataset reflects the behavior of communicating agents for a real-world use case and can be used to learn this.
2. The impact of a corrupted agent's behavior on the overall system is analyzed.
3. Two architectures – centralized and distributed – for detecting anomalies in Multi-Agent System (MAS) communication are compared and evaluated in a simulative study.
4. The impact of use-case specific information availability on detecting anomalies is also considered.

The rest of this paper is structured as follows: In section 2, an overview of approaches considering the impact of anomalies and different anomaly detection approaches is given. Next, section 3 outlines the consideration of anomalies. In section 4, the anomaly detection concept is discussed. An evaluation of the anomaly detection is given in section 5, followed by a conclusion including an outlook in section 6.

2 RELATED WORK

This section provides an overview of related work investigating the impact of anomalies on existing systems and centralized and distributed anomaly detection architectures.

2.1 Analyzing the Impact of Anomalies

Regarding CPES, few approaches consider the impact of attacks on the system (Afrin and Ardakanian, 2023; Zografopoulos et al., 2023). However, none of the approaches focuses on investigating the actual effect of a manipulated agent on the overall system. The investigation of a distributed and centralized architecture for detecting these anomalies is also not considered in the papers mentioned above.

2.2 Comparing Centralized and Distributed Architectures for Anomaly Detection

There are only a few approaches that examine different architectures and none that concurrently investigate the aspect of information availability. Haehner et al. (2013) discuss different architectural concepts for anomaly detection in CPES using Organic

Computing and differentiate local and cooperative anomaly detection. Erhan et al. (2021) give an overview of anomaly detection in sensor systems while also discussing different architectures, such as anomaly detection in the cloud (centralized), in the fog, where information is processed intermediately (between fully decentralized and fully centralized anomaly detection) and anomaly detection at the edge, where the option for a distributed or a collaborative, decentralized computation exists. Furthermore, hybrid anomaly detection models exist, where different architecture models are used in combination (Erhan et al., 2021). However, the authors do not implement anomaly detection, nor do they compare centralized and distributed architectures under consideration of the availability of information.

Centralized Anomaly Detection. Centrally located anomaly detection approaches consider data of the entire system under observation. For this, various approaches have been presented. Turowski et al. (2022) consider electrical loads. Others consider datasets from smart meter data, as Fu et al. (2022), and Farzad and Gulliver (2020) detect anomalies in log messages.

However, none of the approaches considers the comparison of the centralized architecture with others. The aspect of information availability, another contribution of the work at hand, is not discussed. Furthermore, in the approaches that consider information from distributed instances, as electrical loads from different customers by Turowski et al. (2022) or meter data from multiple buildings by Fu et al. (2022), full access to all data is assumed. This might not be possible for all data in CPES, e.g., due to privacy issues. For this reason, information availability is investigated in this paper.

Distributed Anomaly Detection Other approaches to detect anomalies in CPES are implemented in a distributed way: Albarakati et al. (2022) consider a MAS for fault location and cyber attack detection in smart grid applications. Gupta et al. (2022) implement distributed anomaly identification in microgrids and Jithish et al. (2023) consider distributed anomaly detection in smart grids.

Regarding network data, various approaches for distributed anomaly detection exist, as presented by Pei et al. (2022) and Protogerou et al. (2021).

Again, in terms of distributed anomaly detection, none of the approaches carry out a comparison to a centralized architecture while considering the impact of information availability.

Therefore, to the best of our knowledge, none of the existing approaches compares a centralized and distributed architecture for anomaly detection considering information availability. This paper contributes by performing these analyses, considering a MAS for controlling Distributed Energy Resources (DER) in a CPES. The anomalies are caused by manipulating the agents' behavior, which also accounts for the impact of these anomalies on the overall system.

3 COMMUNICATION ANOMALIES IN AGENT-BASED SYSTEMS

This section presents the examined agent system and the induced anomalies. To consider the impact of anomalies on the overall system, the manipulation of agent(s) is necessary to cause anomalies. Effects on the behavior of other agents can only be analyzed if individual agents are manipulated in a simulative environment. This way, the consequences of anomalies on the complete system can be considered. Any consequences can arise long after the actual manipulation, e.g., as reactions to previous anomalies. In this way, the manipulations significantly impact the system as a whole. In this work, an exemplary battery management application is chosen to investigate the effect of manipulated agents in such a system. In this application, individual Battery Energy Storage Systems (BESS) are controlled by a MAS to enable multi-purpose use of these. The agents perform the task of scheduling these devices. The multi-purpose use and the flexibility calculation of the storage systems have been implemented following Tiemann et al. (2022). Our implementation of the BESS management control system has been deployed to the field in an industry project. As a preliminary step before deployment, a hybrid laboratory setup, including simulation and field appliances (industrial Raspberry Pi / Revolution Pi), is used in this work to induce and analyze anomalies in the system. Thus, the simulation scenario comprises the full field setup, including the BESS as found in the field. We focus on the communication between the agents. Considering existing commitments and load forecasts, agents can detect possible scheduling problems. In order to compensate for these problems, the agents can communicate with each other. For this case, the Lightweight Power Exchange Protocol is used, based on the approach of Veith et al. (2014), in which the agents follow a *Four-Way Handshake*. Each agent contains a local power balance solver to solve power imbalances. The im-

plementation of the solver is based on Veith and Steinbach (2017), including the adaptations from Frost et al. (2020). With these extensions, after a given time, the agent determines the possible amount of power at the given time as a solution. The source code is made publicly available ¹.

Whenever an agent detects a scheduling problem, it sends a Notification to its neighbors. The neighboring agents check whether they can solve the problem by providing power, considering their local load forecasts. If it is not completely possible to solve the problem, the missing part of the request is forwarded to the agents' neighbors. This way, the message is forwarded through the complete network of agents. The agents furthermore respond. Little by little, the agent receives offers from the others and tries to solve the problem. The agent has then determined a solution at some point. Subsequently, it informs all agents that are part of the solution. The respective agents check if they can still fulfill the previously offered power (using their local forecasts and commitments). If that is possible, they respond to the requesting agent, which completes the process.

3.1 Consideration of Anomalies

In the following, the consideration of anomalies in an agent-based system in CPES is described. In the setting mentioned above, different agents were manipulated in different ways to induce different types of anomalies: anomalies in the values in the exchanged messages, anomalies in the communication topology, and anomalies in the agents' behavior. In the following, we focus on anomalies in communication behavior to mimic attacks, such as Denial-of-Service attacks. All datasets have been published for further studies and traceability ².

Anomalies in agent behavior affect the system as other agents react to those. To generate anomalies in the agent's behavior, an agent is manipulated to change the behavior of how messages are sent. Since a message starting a negotiation implies many other messages, this message type is chosen. The corresponding agent is manipulated to regularly initiate negotiations in addition to those initiated due to the agents' calculations. Thus, these are started even though no planning problem exists. The agent asks other agents to give or take power, even though there is no reason to do so. As the other agents respond and provide flexibility, this impacts the overall system. The committed power will also be considered in

¹https://github.com/OFFIS-DAI/mango-library/tree/Integration_of_the_LPEP

²<https://zenodo.org/records/7934270>

future calculations and can thus lead to further imbalances. The anomalous negotiation requests are always sent with 50% of the maximum value of the maximum available flexibility. The time interval for which negotiations are started (specified in seconds in the future) and the frequency of negotiation starts are varied. Regarding the anomalies in the behavior, two exemplary datasets are chosen to be discussed. Frequencies for the anomalous negotiation starts every 1 and 15 minutes, 2012 and 8996 seconds in the future, are considered. Since the anomalies affect the system's behavior (other agents forward manipulated messages, etc.), there is a different amount of anomalies in the resulting datasets.

The periods considered cover an average of 18 days. In total, three datasets are considered in this paper: the dataset without anomalies, a dataset containing minutely occurring anomalous negotiation requests, and a dataset containing anomalous negotiation requests sent every 15 minutes. An overview of the three datasets is displayed in the following.

- 0: Data without anomalies
- 1: Anomalous negotiation requests every minute
- 2: Anomalous negotiation requests every 15 minutes

The anomaly-free dataset contains 1.211.851 exchanged messages and 92.143 negotiation starts. The datasets containing the anomalies are discussed in subsection 3.2.

3.2 Impact of Compromised Agents on Communication Behavior

The datasets with the anomalies are discussed in detail to analyze the impact of a corrupted agent on communication behavior in negotiations. When an agent receives and forwards an anomalous negotiation request, this message is also labeled as an anomaly in the dataset. The manipulated agent also sends the anomalous negotiation request to all its neighbors, so more than one anomalous message is already coming from one anomalous negotiation start. Furthermore, when an agent starts an anomalous negotiation request, other agents respond to this. These messages are not labeled as anomalies, as replying to negotiation requests is the correct behavior of other agents. These messages would not have been sent in the absence of the anomalous negotiation requests. Therefore, they can be considered to analyze the impact of the manipulated agent on the system. Messages caused by the negotiation request are responses to the negotiation request, acceptances by the requesting agent, acknowledgments of these acceptances, and

any forwarded messages. The additional messages occurring, as a result, are shown per data record in Table 1. The total number gives the total number of messages caused by the anomalies. The percentage is calculated considering the messages which are not caused by anomalies. For the anomalous negotiation requests every minute, the number of messages doubled (increased to 205,5%), for anomalies occurring every 15 minutes, the number increased to 127,5%.

Table 1: Additional Messages Caused by Anomalous Negotiation Requests.

Dataset	Additional	Total Number
1 minute	41.721	82.604 (205.5%)
15 minutes	8.075	22.918 (123.7%)

For the dataset with anomalies every minute, over 50% of the messages and 19% for 15-minute anomalies would not have been exchanged. These differences can be explained by the frequency of anomalous negotiation starts: the more frequently anomalous negotiation requests are sent, the more responses exist.

The large impact of a manipulated agent on the communication of the overall system can be seen from the number of additional messages.

Another consequence of an anomalous negotiation request is that if the negotiation is successful, other agents will reserve power that is not really needed. This can lead to problems according to the use case. Furthermore, it can occur that agents can no longer meet their obligations due to the (unnecessary) committed and reserved power. Therefore, other agents would start negotiations themselves. This in turn leads to a further increase in communication, which would not exist without the anomalous start of negotiations. These cases were also found sporadically in the datasets.

In summary, agents significantly impact the overall system's communication behavior if they are corrupted in their negotiation behavior. A significant effect on the behavior of the other agents and the communication in the system can be seen.

4 DETECTING ANOMALIES IN AGENT-BASED SYSTEMS

In the following, we discuss a concept to detect the previously described anomalies in such a distributed system. Two architectures for anomaly detection are compared: centralized and distributed. These architectures are displayed in Figure 1. In centralized anomaly detection, shown in 1a, the entire system

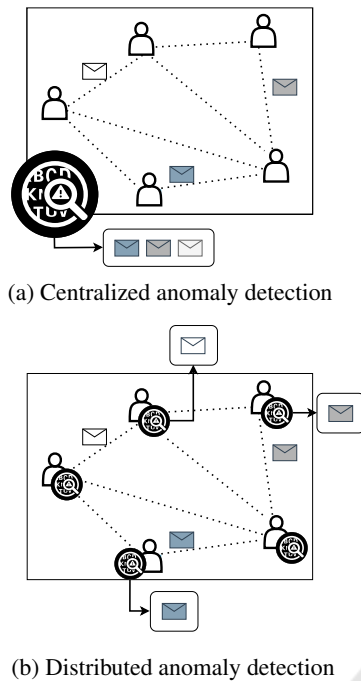


Figure 1: A centralized and a distributed architecture for anomaly detection in agent systems.

is examined at once, and thus, the exchanged messages and information of all agents are considered. In the distributed approach, the anomaly detection model would be implemented for each agent individually, depicted in 1b. This way, only the information regarding the respective agent is considered. In this work, only for the manipulated agent.

The assumption is made that different nodes have access to different information, depending on the use case. It is assumed that in some cases, information regarding other agents or DER is only available locally. Due to reasons such as privacy, data protection, data exchange minimization, or regulatory restrictions, this information may not be exchanged. Therefore, in some cases, distributed anomaly detection may be required since each agent has individual information that is not shared with a centrally observing anomaly detection system. Centralized anomaly detection could be installed if information about the complete system and the messages exchanged are available, as well as information about each agent in detail. For the considered use case of distributed control of DER, the information is either distributed at each agent or available at a central location. Therefore, no hybrid architecture is investigated.

The impact of information availability on anomaly detection performance is additionally investigated. It is assumed that different levels are to be examined, having different information available. It is assumed

here that, in summary, two levels of information availability regarding the exchanged messages can be identified.

1. Sending agent and timestamp
2. Message content

On the first level, no insight into the message is given. These could be, e.g., encrypted and thus not accessible. This level only considers the agent which sends the message and the timestamp of each message. The second level considers the content of the messages, as exchanged information. Therefore, on this layer, insight into the messages is given.

The two layers are thus divided into specific (limited) metadata about the messages (when each message was sent and by which instance) and the concrete message content. This shows analogies to anomaly detection in IP networks. There, either only the header can be inspected (e.g. if the content of the message is encrypted) or the content of the packets can be inspected (in Deep Packet Inspection). The impact of information availability on the anomaly detection performance is investigated in this paper, considering the two different layers.

The following information is provided per level when applied to the present setting using the power exchange protocol.

- 1. Sending agent and timestamp: Only the information about which agent sends which message at which time is given.
- 2. Message content: Insight into the exchanged negotiation messages is given. The message content contains, for example, the message type (e.g., *Demand Notification*) and the power value.

The data for anomaly detection consists of the agents' exchanged messages. The results are based on the same datasets in order to establish comparability. For the distributed anomaly detection architecture, the datasets are adjusted to consider only the messages sent by one agent: the manipulated one. The result is a reduction in the size of the datasets and an increase in the percentage of anomalies. Since anomalies are outliers, the proportion of anomalies should not be too large to not preclude comparability with other anomaly detection methods and data. For this reason, the datasets for distributed anomaly detection are adjusted and filled every second. Thus, it is mimicked that the data is stored regularly as time-series-based data. This furthermore allows comparison with other time-based models and datasets.

5 EVALUATION

This section describes the results of the anomaly detection, considering the different approaches: Isolation Forest (IF), Support Vector Machine (SVM), autoencoder (AE), and Graph-Deviation Network (GDN). The selected approaches for anomaly detection are chosen based on their high performances in several applications. Isolation Forests perform well in detecting anomalies in log messages (Farzad and Gulliver, 2020). SVMs are used due to their performance in detecting attacks in smart grids, as in Niu et al. (2019). Autoencoders detect anomalies well in critical infrastructures, as listed by Mavikumbure et al. (2022). Furthermore, a graph-based approach is implemented, in which the system under consideration is interpreted as a graph, which applies accordingly to the communication topology of the agents. The GDN from Deng and Hooi (2021) is used, which detects anomalies well in several use cases (Chen et al., 2021). The implemented models can be found in GitLab, including selected parameters³.

Results show that the specificity was always similar (0.98-1.0), which means that the models predict the negative instances very well. For this reason, the metrics referring to anomalous entries are considered for the discussion: recall (ability to find actual anomalous entries), precision (correctly predicted positive entries) and the F1 score (combination of the metrics).

5.1 Centralized Anomaly Detection

In the following, the results of the centralized anomaly detection are discussed. All centralized approaches achieve the best results considering agent and timestamp. Accordingly, insight into the messages does not lead to improvements. For the dataset with anomalous negotiation requests every minute, the results are overall very poor, with recall not over 0.55 for all approaches, as shown in Table 2. The results are better overall for the anomalous negotiation requests every 15 minutes. For the Isolation Forest, the results are still not very good: a precision of 0.23. The SVM achieves similar, slightly better results and the autoencoder achieves even better results. However, the autoencoder only achieves a recall from 0.55. The GDN detects the anomalies the best, with all metrics above 0.89.

In summary, the 15-minute anomalies are better recognizable by the centralized anomaly detection, while the 1-minute anomalies are not recognizable. The GDN performs best.

³<https://gitlab.com/digitalized-energy-systems/models/anomaly-detection-in-cpes>

Table 2: Communication anomalies: Centralized anomaly detection.

Frequency	Model	Precision	Recall	F1
1 Minute	IF	0.3	0.55	0.37
	SVM	1.0	0.08	0.04
	AE	1.0	0.02	0.04
	GDN	0.5	0.5	0.5
15 Minutes	IF	0.23	0.89	0.40
	SVM	0.71	0.57	0.63
	AE	1.0	0.55	0.71
	GDN	0.89	0.97	0.93

Table 3: Communication anomalies: Distributed anomaly detection.

Frequency	Model	Precision	Recall	F1
1 Minute	IF	0.79	1.0	0.86
	SVM	0.82	0.78	0.80
	AE	0.79	1.0	0.89
	GDN	0.92	0.53	0.67
15 Minute	IF	0.18	1.0	0.31
	SVM	0.46	1.0	0.6
	AE	0.29	1.0	0.44
	GDN	0.92	0.21	0.34

5.2 Distributed Anomaly Detection

The results of the distributed anomaly detection approaches are presented in the following, as shown in Table 3. Regarding the dataset with anomalous negotiation starts every minute, the approaches achieve different performances, with precision values of 0.76 - 0.92, recall of 0.53 - 1.0, F1 scores of 0.67 and 0.89. For the anomalous negotiation requests occurring every 15 minutes, the SVM performs best, but the performance is insufficient overall (precision below 0.5).

Regarding the information used, the Isolation Forest and autoencoder achieve the best results considering timestamp, message type, sender, neighbor and receiver. The GDN achieves the best results considering the timestamp, message type, sender and neighbor. The SVM achieves similar results without considering the message content; therefore, considering the timestamp and the agent is sufficient.

In the distributed anomaly detection, anomalies in 15-minute intervals are barely discernible, whereas 1-minute intervals are better recognizable. The autoencoder performs best for the 1-minute intervals.

5.3 Discussion

The centralized approaches are better at detecting anomalous negotiation requests started with a fre-

quency of 15 minutes than those started every minute. The GDN achieves the best results in the centralized approach, but only good results for the anomalous negotiation requests occurring every 15 minutes. The reverse is valid for the distributed anomaly detection: anomalous 1-minute intervals can be detected better than 15-minute intervals. The only approach that performs well in detecting the anomalies for the distributed architecture is the autoencoder, but only regarding anomalous negotiation starts occurring every minute. Therefore, none of the architectures under consideration are superior to the other ones at identifying communication anomalies.

The fact that different frequencies of anomalous negotiation requests can be detected better or worse by different architectures can be explained by the irregular occurrence of negotiations in the normal setting. The centralized anomaly detection recognizes patterns in the entire system better as the data is available for all agents, and thus, information about the irregular starting negotiations is learned. This way, negotiations starting anomalously every 15 minutes can be recognized better. Distributed anomaly detection has only data for one agent available, making drawing conclusions about the entire system's behavior challenging. In normal behavior, without manipulated agents, negotiations are not triggered regularly but whenever a scheduling problem is detected. Thus, it also appears in the normal data that no negotiation is triggered for a long time. Therefore, it cannot satisfactorily detect anomalies occurring every 15 minutes. On the other hand, the anomalous negotiation requests every minute are easier to detect. An agent that does not often begin negotiations in the normal data was chosen. Since the distributed anomaly detection focuses on this agent, this is better detectable. The centralized anomaly detection is not able to detect the minute-by-minute anomalous negotiation requests because, in the overall system, it occurs more often that negotiations take place very frequently since all agents start negotiations.

6 CONCLUSION

To investigate the effect of manipulated agents on MAS in CPES, we presented a dataset in which an agent was manipulated to send anomalous messages to others. The significant impact of this agent on the communication of the MAS was shown. Furthermore, we implemented anomaly detection approaches to detect these anomalous messages. For this purpose, two architectures have been applied: centralized and distributed. Different models were implemented: Iso-

lation Forest, SVM, autoencoder, GDN. The results show that the chosen architecture significantly impacts the performance. No architecture performs best for all anomalies. The autoencoder and GDN predominantly achieve the best results, followed by the SVM.

To discuss the results based on the information given, different levels of information availability are considered.

1. Agent and Timestamp of the Message. At this level, no insight into the messages is given, only agent and timestamp are considered. For the centralized anomaly detection, adding the content of the message did not improve the results. For the distributed anomaly detection, the performance of the SVM did not improve when adding the message content.

2. Content of the Message. More insight is given if the content of the messages is accessible. The autoencoder and GDN of the distributed architecture achieved slightly better, the Isolation Forest much better results when considering the content of the messages. For the centralized anomaly detection, insight into the messages did not lead to improvements.

It is not possible to make a clear statement regarding the architecture to select, since different results exist for the anomalies in the communication behavior depending on the frequency. Since the architecture has a significant impact on the performance of the anomaly detection though, this information should be considered when designing an observer for anomaly detection in agent systems.

For the centralized architecture, the insight into the messages does not change the performance. This is an advantage since the insight is not always given in reality, which should be taken into account to reflect privacy concerns. The results show that implementing an anomaly detection observer in distributed systems is a significant challenge.

Current and future work include the following topics: The potential change in the anomaly detection approaches with additional types of anomalies, for example, anomalies in the communication topology of agents. Furthermore, to improve the performance, a combined detection, e.g., using ensemble learning, could be analyzed. Additionally, the presented approaches can be extended by integrating concepts of trust from Organic Computing into the existing agent system. To improve the robustness of a given system, interactions can be limited to trustworthy agents (Klejnowski et al., 2010).

REFERENCES

- Afrin, A. and Ardakanian, O. (2023). Adversarial attacks on machine learning-based state estimation in power distribution systems. In *Proceedings of the 14th ACM International Conference on Future Energy Systems*, pages 446–458.
- Albarakati, J., Azeroual, M., Boujoudar, Y., EL Iysaouy, L., Aljarbouh, A., Tassaddiq, A., and EL Markhi, H. (2022). Multi-agent-based fault location and cyber-attack detection in distribution system. *Energies*, 16(1):224.
- Chen, P.-Y., Cheng, S.-M., and Chen, K.-C. (2012). Smart attacks in smart grid communication networks. *IEEE Communications Magazine*, 50(8):24–29.
- Chen, Z., Chen, D., Zhang, X., Yuan, Z., and Cheng, X. (2021). Learning graph structures with transformer for multivariate time-series anomaly detection in iot. *IEEE Internet of Things Journal*, 9(12):9179–9189.
- Deng, A. and Hooi, B. (2021). Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4027–4035.
- Erhan, L., Ndubuaku, M., Di Mauro, M., Song, W., Chen, M., Fortino, G., Bagdasar, O., and Liotta, A. (2021). Smart anomaly detection in sensor systems: A multi-perspective review. *Information Fusion*, 67:64–79.
- Farzad, A. and Gulliver, T. A. (2020). Unsupervised log message anomaly detection. *ICT Express*, 6(3):229–237.
- Ferragut, E. M., Laska, J., Czejdo, B., and Melin, A. (2013). Addressing the challenges of anomaly detection for cyber physical energy grid systems. In *Proceedings of the eighth annual cyber security and information intelligence research workshop*, pages 1–4.
- Frost, E., Veith, E. M., and Fischer, L. (2020). Robust and deterministic scheduling of power grid actors. In *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, volume 1, pages 100–105.
- Fu, C., Arjunan, P., and Miller, C. (2022). Trimming outliers using trees: winning solution of the large-scale energy anomaly detection (lead) competition. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 456–461.
- Gupta, K., Sahoo, S., Mohanty, R., Panigrahi, B. K., and Blaabjerg, F. (2022). Decentralized anomaly identification in cyber-physical dc microgrids. In *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1–6.
- Haehner, J., Rudolph, S., Tomforde, S., Fisch, D., Sick, B., Kopal, N., and Wacker, A. (2013). A concept for securing cyber-physical systems with organic computing techniques. In *26th International Conference on Architecture of Computing Systems 2013*, pages 1–13.
- Jithish, J., Alangot, B., Mahalingam, N., and Yeo, K. S. (2023). Distributed anomaly detection in smart grids: A federated learning-based approach. *IEEE Access*, 11:7157–7179.
- Klejnowski, L., Bernard, Y., Hahner, J., and Müller-Schloer, C. (2010). An architecture for trust-adaptive agents. In *2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshop*, pages 178–183.
- Mavikumbure, H. S., Wickramasinghe, C. S., Marino, D. L., Cobilean, V., and Manic, M. (2022). Anomaly detection in critical-infrastructures using autoencoders: A survey. In *IECON 2022—48th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–7. IEEE.
- Nieße, A. and Tröschel, M. (2016). Controlled self-organization in smart grids. In *2016 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–6. IEEE.
- Niu, X., Li, J., Sun, J., and Tomsovic, K. (2019). Dynamic detection of false data injection attack in smart grid using deep learning. In *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–6.
- Pei, J., Zhong, K., Jan, M. A., and Li, J. (2022). Personalized federated learning framework for network traffic anomaly detection. *Computer Networks*, 209:108906.
- Protogerou, A., Papadopoulos, S., Drosou, A., Tzouvaras, D., and Refanidis, I. (2021). A graph neural network method for distributed anomaly detection in iot. *Evolving Systems*, 12(1):19–36.
- Schmeck, H., Müller-Schloer, C., Çakar, E., Mnif, M., and Richter, U. (2010). Adaptivity and self-organization in organic computing systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 5(3):1–32.
- Tiemann, P. H., Nebel-Wenner, M., Holly, S., Frost, E., Jimenez Martinez, A., and Nieße, A. (2022). Operational flexibility for multi-purpose usage of pooled battery storage systems. *Energy Informatics*, 5(1):1–13.
- Turowski, M., Heidrich, B., Phipps, K., Schmieder, K., Neumann, O., Mikut, R., and Hagenmeyer, V. (2022). Enhancing anomaly detection methods for energy time series using latent space data representations. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, pages 208–227.
- Veith, E., Steinbach, B., and Windeln, J. (2014). A lightweight distributed software agent for automatic demand—supply calculation in smart grids. *International Journal On Advances in Internet Technology*, 7(1):97–113.
- Veith, E. M. S. P. and Steinbach, B. (2017). Agent-based power equilibrium in a smart grid with xboole. In *2017 International Conference on Information and Digital Technologies (IDT)*, pages 406–416.
- Zografopoulos, I., Hatzirygiouri, N. D., and Konstantinou, C. (2023). Distributed energy resources cybersecurity outlook: Vulnerabilities, attacks, impacts, and mitigations. *IEEE Systems Journal*.