International Journal of Information Security (2026) 25:161
https://doi.org/10.1007/s10207-026-01335-w

REGULAR CONTRIBUTION

Beaconing detection via timing-based flow metadata analysis: implications for SCADA/ICS

Mustafa Tahsin Yilmaz[1,2] · Ferdi Doğan[3] · Esra Söğüt[4] · Enes Algül[5] · Çağla Aksoy[6]

Received: 20 June 2026 / Accepted: 8 September 2026 / Published online: 16 September 2026
© The Author(s) 2026

Abstract

Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems (ICS) are critical to the continuity of modern industrial infrastructure. Cyberattacks targeting these systems can lead to serious consequences, including production losses, physical damage, and threats to human safety. In particular, detecting covert Command and Control (C2) beacon communications within encrypted network traffic poses a significant security challenge because the proposed method operates exclusively on flow-level timing metadata without payload inspection, making it inherently applicable to both plaintext and TLS-encrypted traffic. In this experimental study, a lightweight, hybrid beacon detection method is proposed that uses only connection metadata generated by Zeek, without requiring deep packet inspection or labeled datasets. The proposed approach combines Coefficient of Variation (CV)-based statistical periodicity analysis with an unsupervised anomaly detection method based on Isolation Forest to identify periodic communication patterns with low jitter that are characteristic of beacon behavior. As part of the experimental study, an isolated SCADA-like laboratory environment was provided in which web traffic mimicking human behavior and periodic beacon traffic were generated simultaneously. The hybrid detection process was applied by analyzing connection times, connection intervals, average duration, standard deviation, and periodicity metrics obtained from Zeek "conn.log" records. Additionally, the proposed method was evaluated in comparison with Long Short-Term Memory (LSTM) and Random Forest (RF) algorithms. Experimental results demonstrated that the proposed hybrid approach achieved a detection rate (recall) of 1.00 and a false positive rate of 0.056 (Precision=0.33, F1-Score=0.50) at the connection-group level, consistently ranking the true beacon first (anomaly score = 1.00) among all 37 evaluated connection groups and exhibited higher performance compared to LSTM and RF models. The findings reveal that metadata-based anomaly analysis can effectively detect beacon communications in encrypted SCADA/ICS networks without requiring content visibility. The method proposed in this study offers an effective solution for industrial network security in terms of scalability, explainability, low computational cost, and real-time applicability.

Keywords SCADA security · Industrial control systems · Beacon detection · Timing-based traffic analysis · Unsupervised anomaly detection · Network security · Command-and-control detection

✉ Enes Algül
ealgul@bingol.edu.tr

Mustafa Tahsin Yilmaz
myilmaz@kau.edu.sa

Ferdi Doğan
fdogan@adiyaman.edu.tr

Esra Söğüt
esrasogut@gazi.edu.tr

Çağla Aksoy
cagla.aksoy@msu.edu.tr

[1] Department of Industrial Engineering, Faculty of Engineering, King Abdulaziz University, 21589 Jeddah, Saudi Arabia

[2] Center of Research Excellence in Artificial Intelligence and Data Science (AIADS), King Abdulaziz University, Jeddah, Saudi Arabia

[3] Department of Computer Engineering, Adıyaman University, Adıyaman 02040, Turkey

[4] Department of Computer Engineering, Faculty of Technology, Gazi University, Ankara 06560, Turkey

[5] Department of Computer Engineering, Bingöl University, Bingöl 12000, Turkey

[6] Department of Computer Engineering, National Defence University, Ankara 06654, Turkey

Abbreviations

CNN	Convolutional neural networks
CV	Coefficient of variation
C2	Command-and-control
HMI	Human-machine interface
ICS	Industrial control systems
iForest	Isolation forest
LSTM	Long short-term memory
PLC	Programmable logic controller
RTU	Remote terminal unit
SCADA	Supervisory control and data acquisition

1 Introduction

Industrial systems involve processes where new approaches are being adopted alongside the rapid advancement of technology. One such approach is Supervisory Control and Data Acquisition (SCADA) systems. These systems enable the control of various devices within industrial systems from a single point. Device monitoring and reporting of results are provided through central control. SCADA systems are used in large-scale industrial production, infrastructure, and manufacturing facilities. Examples include the energy sector, water and wastewater management facilities, oil and natural gas facilities, and production facilities (Figure 1).

SCADA systems include various sensors, RTUs (Remote Terminal Units), PLCs (Programmable Logic Controllers), actuators, HMIs (Human-Machine Interfaces), and central management software (Figure 2) [1–3]. These systems contain numerous devices, which are managed from a central location. A network environment exists to retrieve data from the devices. This network environment is implemented by data flow providers such as wired connections, serial ports, radio frequencies, and GSM. All elements in the system are monitored from a single center, malfunctions are checked, and necessary interventions are attempted [4, 5].

SCADA systems experience heavy data traffic, with the central control unit continuously displaying incoming data, sending commands, and managing work processes. A large portion of such networks have an open architecture. This means that the system's control processes can be carried out not only at the industrial center but also in all environments connected to the internet by establishing the necessary connections. These externally open systems require data security [6]. Various attacks from outside or inside can disrupt the system's operation. This situation can cause irreparable physical damage and lead to consequences that could put human lives at risk [7]. For this reason, data security in SCADA systems is an issue that requires careful attention.

In SCADA systems, network traffic must be monitored as it flows to the central management point. Detecting the presence of cyber threats early and reliably during central control processes is challenging in terms of ensuring security. The use of TLS/HTTPS encryption in network traffic largely renders content-based monitoring ineffective. Therefore, there is an increasing need for methods that utilize metadata [8]. Malicious software such as Beacon, which mimics legitimate system behavior by transmitting data to the central control system at fixed intervals, poses a serious threat to SCADA networks [9]. Such software is used to covertly monitor devices or central management, making early detection essential for maintaining proper system operation.

The main objective of this study is to develop a lightweight and effective method to monitor data in network traffic and detect beacon attacks in a non-surveillance-independent and content-independent manner, without interfering with metadata in network traffic. The study combines periodicity analysis via connection metadata with an untagged machine learning approach. It is evaluated on SCADA-based network traffic generated in a laboratory environment. The focus is on determining low-deviation command and control communication based on metadata. With these approaches, the study aims to address gaps in the literature.

1.1 Problem definition and research rationale

Sensors or devices in SCADA systems exhibit specific behaviors on the network. Beacon attacks mimic legitimate device behavior to evade detection in these networks. Beaconing attacks send signals to Command-and-Control (C2) points at specific intervals, just like a device on the network [10]. The system may perceive such situations as normal network traffic. Data in network traffic is transmitted in encrypted packets using protocols such as TLS/HTTPS. This renders content inspection of the data on the network ineffective, making information about the packet invisible [11].

In SCADA systems, network data with similar structures within regular traffic processes appear normal in the system as Beacon behaviors. Beacon attacks often overlap with standard traffic [11]. Small time deviations (jitter) in data can sometimes be observed in the network during maintenance activities and shift changes. In such cases, inter-arrival distributions, flow directions, flow durations, packet statistics, and endpoint habits must be evaluated together [12]. Network traffic predominantly consists of normal behavior, with labeled beacon traffic representing only a small fraction, making detection challenging. This makes it difficult to detect attacks using machine learning and artificial intelligence (AI) methods. Obtaining network traffic data with beacon behavior is quite difficult in this context [13].

The protection of SCADA systems is critically important given the severity and frequency of emerging cyber threats.

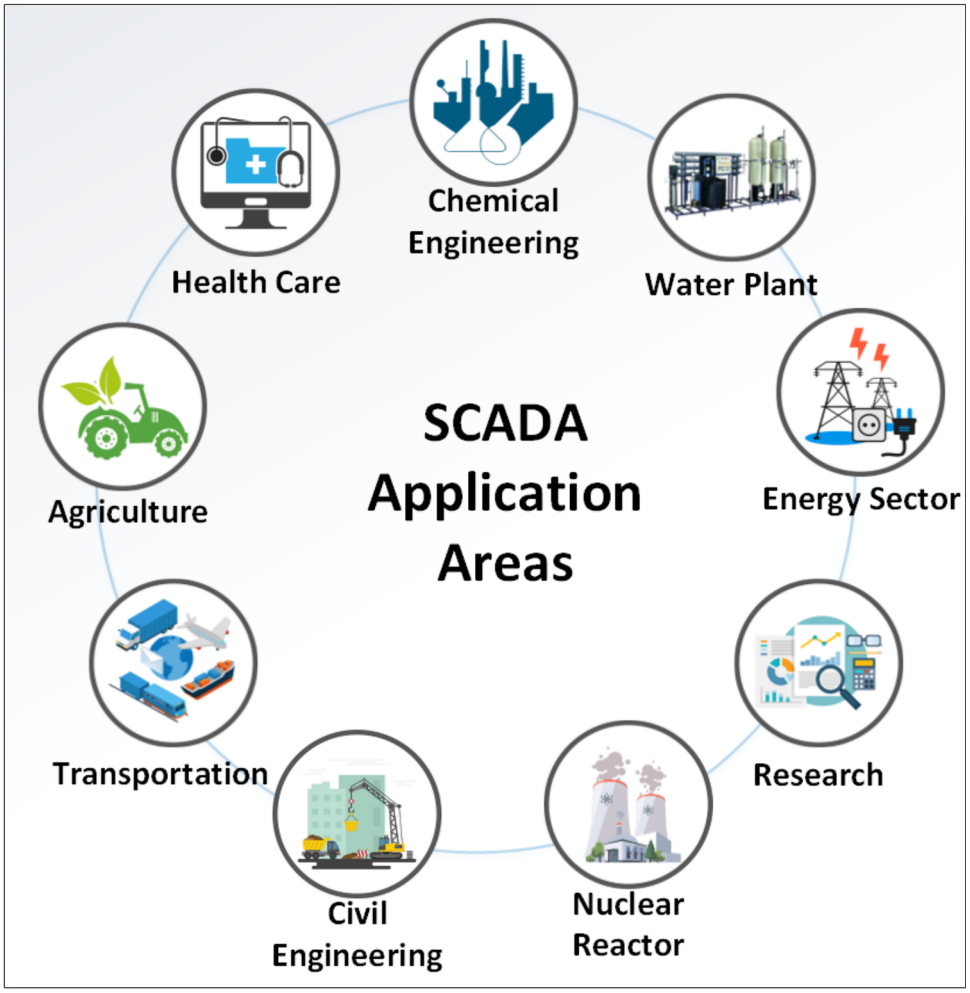


Fig. 1 Examples of SCADA system applications

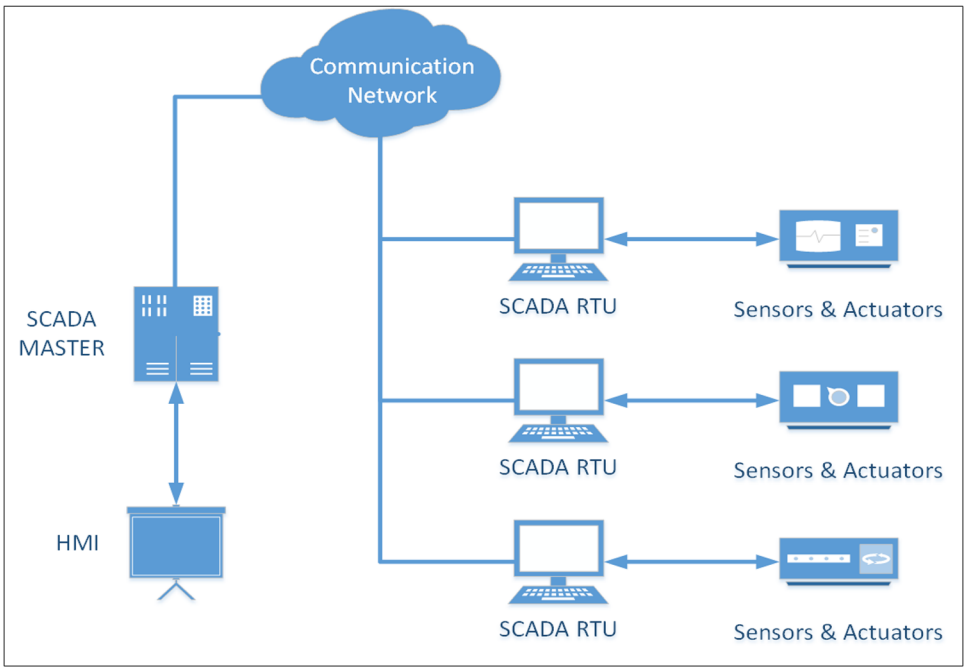


Fig. 2 SCADA system architecture

Examples of these include Stuxnet Worm [14], Schneider Triconex Tricon [15], and attacks on the energy sector [16]. Stuxnet was an attack targeting nuclear facilities and causing physical damage. The Schneider Triconex Tricon attack manipulated security protocols, causing system malfunctions. The attacks targeting the energy sector in 2013 demonstrate the high level of cyber threats. These attacks exhibit a destructive approach and indicate the involvement of long-term control and monitoring processes. Early detection and preventative measures against these attacks are essential.

This study utilizes metadata extracted by the Zeek tool to establish an anomaly-focused detection pipeline. Data obtained from Zeek can provide rich data with attributes such as duration, direction, orig/resp bytes and packet counts, connection status, retry indicators, and, if available, SNI/certificate metadata [17]. Providing such data richness enables the acquisition of regular but unusual periodic flow information. Additionally, it distinguishes low-bandwidth and persistent C2 communication despite the natural processes of network traffic [18]. The main reasons for this research can be listed as follows:

- Security and rapid detection: Late detection of attacks in SCADA systems leads to production loss, physical damage, and risks to human safety. Rapid and early detection of beacon attacks is critical in this regard.
- Adaptability to the field: There is a need for an approach that does not require deep inspection of data packets in network traffic in SCADA systems and remains effective regardless of whether traffic is encrypted.
- There is also a need for an architecture that can operate with low load on devices on the network. This enables sustainability for Industrial Control Systems (ICS) with limited resources.
- Generalizability: Reducing the dependency on tagged data in Beacon attacks, which are rare in the dense network traffic generated by such systems, is an important step. An anomaly-based approach that can quickly adapt to such behaviors seen in various industrial systems can increase the possibility of generalization to different sectors.

The primary objective of this study is not to access periodic Beacon packets in network traffic. It is to design metadata that can distinguish between intrusion attempts in systems with industrial differences. This is to be achieved in near real-time and with minimal load on the system. The fundamental concept of the study is to present a method that detects Beacon behavior in systems such as ICS and SCADA, despite encrypted networks. To this end, it utilizes a combination of Zeek-based feature extraction and anomaly detection (Isolation Forest).

1.2 Research gap and contributions of the study

Numerous studies have been conducted on intrusion detection and anomaly analysis in SCADA and ICS networks. The literature contains a wealth of work in this field. However, studies on detecting beaconing behavior in encrypted industrial network traffic are insufficient. A large portion of the existing studies focus on content analysis and signature-based analyses. Many studies are based on supervised learning methods using labeled datasets. [19–21]. The fact that network traffic in real industrial environments is encrypted, that content access is not always possible, and that data labeling involves time and difficulty, limits the applicability of these methods. Studies that work with periodic communication patterns like beaconing and metadata, and that do not require labeling, are quite limited. Therefore, a significant research gap is observed in SCADA and ICS environments requiring real-time monitoring.

This study proposes a hybrid approach to detect beaconing behavior in SCADA and ICS network traffic using Zeek-based communication metadata [11, 22]. Therefore, it aims to fill a gap in the literature. The study combines the analysis of low-deviation periodic communication, such as beacon behavior, through coefficient of variation and timing, with Isolation Forest-based unlabeled anomaly detection. This approach can be considered the unique aspect of the study. This method offers a structure that does not require content analysis, reduces the need for labeled data, and has low computational costs. The study provides a security solution for SCADA and ICS environments that enables Timing-Based Traffic Analysis, is lightweight, explainable, and allows for real-time implementation. Thus, it makes a unique contribution to the literature. Gaps identified in the current literature:

- The presence of encrypted traffic on the network and the unsuitability of Payload approaches for the field: SCADA systems require large amounts of labeled data. The methods used (Random Forest, Long Short-Term Memory, etc.) require a large amount of labeled data. Furthermore, applications are limited due to the inability to analyze encrypted network traffic [23].
- Inadequacy of anomaly detection methods containing temporal data: Regular temporal activities in ICS systems can cause false positives in anomaly detection. There is a need for metadata-based solutions that examine temporal patterns in network traffic and adapt to encrypted traffic [24].
- Limited use of unsupervised learning methods for unlabeled data: Algorithms with unsupervised learning methods eliminate the need for labeled data. However, most studies have ignored periodic behaviors. Such studies have focused solely on statistical variables [25].

Considering the findings revealed by gaps in the literature, the aim of this study is to contribute to the literature within the scope of its objectives. In this context, the contributions of this study to the literature can be summarized as follows:

- Use of metadata using the Zeek tool: This study focused on time-based attributes such as inter-arrival times and change coefficients through metadata. This provides a state-content independent Beacon detection. Thus, Timing-Based Traffic Analysis is performed, and security breaches can be detected early.
- Integration of statistical heuristic filtering with unsupervised anomaly detection: The proposed hybrid method does not merely combine two algorithms but uses the Coefficient of Variation (CV) as a context-aware filter to calibrate the input space for the Isolation Forest. This integration employs CV as an intuitive candidate selection stage prior to Isolation Forest-based anomaly scoring, in addition to combining the two algorithms in the proposed hybrid method. While the CV stage aims to prioritize low-frequency periodic streams within a broader metadata stream, the machine learning stage performs the final multi-feature anomaly assessment.
- Integration significantly reduces the processing of legitimate non-periodic traffic, allowing the machine learning model to focus on subtle anomalies within a high-probability candidate set, thereby enhancing both precision and computational efficiency in resource-constrained SCADA environments.
- Label-free and field-adaptable structure: The study does not require labeled data. It is designed to be compatible with SCADA systems that require low bandwidth and low processing volume, suitable for the field.
- High accuracy, zero false negatives: The proposed method achieves 94.59% accuracy with a sensitivity of 1.0000, ensuring that no beacon activity is missed, the primary safety requirement in SCADA/ICS environments. The 5.6% false positive rate is acknowledged as an operational trade-off of unsupervised anomaly detection in periodic-traffic environments and is addressed through ranked alerting and whitelisting strategies discussed in Section 5.

The study demonstrates that the presented method contributes academically, is applicable in the field, and is suitable for real-time operation. It also introduces a robust early warning system based on encrypted network traffic. This addresses significant gaps in the literature.

2 Related works

This section will examine the primary studies conducted on the detection of Beacon attacks in SCADA and ICS systems. Current methods, along with their advantages and disadvantages, will be presented through literature reviews.

2.1 Traditional anomaly detection in SCADA and ICS networks

Anomaly detection in SCADA and ICS networks is crucial for maintaining system security and integrity. Traditional methods used for anomaly detection in such systems include signature-based methods, rule-based approaches, and statistical methods. Signature-based methods incorporate known attack patterns. These known at-tacks are stored in a database. To identify potential threats in the traffic occurring on the network, this dataset is compared with these signatures. While the recorded frequency of targeted cyberattacks in SCADA and ICS environments remains relatively low compared to IT systems, the specialized nature of these infrastructures often limits the visibility and detection of covert activities [26]. Although traditional signature-based attacks are limited, they refer to the signature-based dataset for attack detection at the packet content level. One study emphasized that combining the use of Bloom filters to store the signature database with other techniques could be effective [27].

Rule-based approaches consist of a set of predefined rules for attack detection. These rules are based on logical rules within the network. Rule-based systems have been noted to be potentially more effective in ICS environments due to their nature [28]. An example of a rule-based system is a rule-based engine that can analyze traffic metadata against baseline data and then trigger an alert system to mitigate threats [29].

Statistical methods model abnormal behavior to identify deviations in the network. This involves extracting statistical data from normal ICS communication models and identifying deviations from this data as potential anomalies. Statistical methods have stable communication patterns that may be more suitable for ICS networks [30, 31]. Although traditional methods are effective in scenarios such as SCADA and ICS environments, they cannot adapt to evolving and changing threats. These types of networks are becoming increasingly complex. Supervised machine learning and AI methods are replacing these methods [32, 33].

Signature-based methods cannot directly see protocol signatures during an attack because the packet content is hidden. This leads to low accuracy in detection systems. Rule-based systems act according to known scenarios. It is difficult to detect attacks hidden by encryption. Since protocol behaviors are hidden, their applicability to predefined rules is weak. Statistical characteristics of packets in network traffic, such

as size and timing, can be altered by encryption. This weakens the accuracy of statistical methods. Furthermore, statistical methods require additional processing power due to the computational intensity involved in large data sets. These are the general limitations of traditional methods. Unlike traditional methods based on deep packet inspection, the threat model addressed in this experimental study focuses on detecting covert attackers aiming to maintain persistent access through periodic C2 signals. The proposed system overcomes the limitations imposed by encrypted protocols such as TLS/HTTPS by using Zeek connection metadata instead of payload analysis. Focusing on behavioral patterns such as timing and periodicity ensures the validity and effectiveness of the detection framework in modern, high-security SCADA environments.

2.2 Machine learning-based SCADA security studies

In machine learning (ML) studies for attack and anomaly detection in SCADA and ICS systems, supervised learning, deep learning, and time series analysis are employed. ML-based studies conducted in recent years have demonstrated over 90% success rates. This highlights the effectiveness of ML and AI studies in SCADA systems [34, 35]. In SCADA security applications, supervised, unsupervised, and reinforcement learning methods focus on anomaly threat detection processes [36–38]. Deep learning (DL) methods such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) and time series analysis are used in these studies [39, 40].

Recent studies in SCADA and ICS environments show that ML-based intrusion detection systems have not only performed supervised classification. In these environments, time series-based DL [41], self-supervised learning [42], graph-based modeling [43] and transformer-based attention mechanisms [44] are prominent approaches [45]. LSTM [46], attention, and transformer-based approaches [47] are preferred for modeling patterns in time-dependent ICS traffic. In addition, graph-based and graph attention-based methods are proposed to learn the relationships between devices, flows, and system components [48–50]. In environments where labeled data is limited, self-supervised and unsupervised detection approaches [51–53] are prominent. Although there are various studies, the high computational cost of the current methods used is seen as a significant problem. In addition, dataset dependency, limited explainability, and adaptability to real field environments are important problems.

SCADA/ICS security studies need to be evaluated from contextual, semantic, and cyber-physical perspectives. Hetu et al. proposed CASI, a context-aware semantic inference framework for combining camera videos with network traffic in such systems. This method aimed to extract the physical semantics of process variables. They used multimodal and non-intrusive analyses to understand them at the semantic level [43]. In another study, Zhang et al. analyzed the vulnerability of load frequency control to network parameter attacks. They showed that manipulation of control systems in a SCADA environment can extend to physical system dynamics and frequency instability [52]. These studies show that research in SCADA/ICS security is increasingly expanding from packet or flow-level detection to context-sensitive interpretation and physical consequences analysis. This study aims to provide a practical and low-cost monitoring layer for SCADA/ICS security monitoring. Furthermore, the proposed method detects weak beaconing signals in encrypted traffic using Zeek connection metadata and Isolation Forest. Table 1 lists some methods used for attack and anomaly detection in SCADA systems, along with the datasets and their purposes.

2.3 Beaconing and periodic communication detection studies in SCADA and ICS networks

Various studies have been conducted on attack detection in SCADA and ICS networks. Among these attacks, Beacon-type attack detection holds a significant place in the literature. Temporal analysis, periodicity mining, and metadata-based Beacon anomaly detection are frequently encountered in the literature. Studies include anomaly detection efforts resistant to time variations using time series processing and LSTM-based models [60]. A study conducted for EtherCAT-based networks performed anomaly detection by making periodic inferences at the device level [61]. A study attempting to combine DL approaches using statistical time series models proposed statistical ML [62, 63]. Several studies have conducted classification studies that could be effective in encrypted traffic using time-frequency feature fusion and periodogram analysis [64]. Vankatesan [65], Apruzzese [66], and Yeh [67] conducted studies on detection based on periodic communication signatures from botnet C2 Beacon behavior in campus networks. Ni et al. [68] and Efiong et al. [69] presented Modbus and Grassmarlin-based solutions. They conducted studies on periodic patterns and metadata extraction in SCADA and ICS networks. Table 2 summarizes these studies, and the important information obtained from them.

While the studies surveyed above demonstrate varied approaches to beaconing and periodic traffic detection, existing tools such as RITA (Real Intelligence Threat Analytics) and Zeek-native heuristics detect beaconing primarily through connection frequency analysis and periodic interval scoring Rather than competing with these tools, the proposed approach is positioned as a complementary data-driven alternative. A simple rule-based proxy (CV < 0.50 AND mean interval $\in [40, 80]$ s) was evaluated as a baseline (Table @10); results confirm that the Isolation Forest

Table 1 Some attack and anomaly detection studies in SCADA systems

Architecture	Description	Dataset
CNN [54]	Used for spatial data processing in power grids	Synthetic supply–demand curves
LSTM [55]	Used for temporal data analysis in Modbus/TCP-based SCADA systems	sWAT dataset
Hybrid CNN-LSTM [56]	A unified model for unauthorized access detection in smart grids	DNP3 dataset
Transformer [54]	Used with CNNs and LSTMs for attack detection in power grids	Synthetic datasets
GAN [57]	GAN-based approach for attack detection and mitigation in power systems	Simulated false data injection and DoS attacks
VAE [58]	VAE framework combined with RNNs and DNNs for covert attack detection	IEEE 68-bus and 123 bus systems
Hybrid GC-LSTM [59]	Used for attack detection in power systems	IEEE 68-bus system
1D CNN [34]	Highest accuracy achieved in anomaly detection in ICS	Generated ICS communication patterns

stage adds value by reducing false positives. Although effective in general IT network environments, these tools rely on fixed interval assumptions, require labeled traffic samples or manual configuration, and do not natively address the low-bandwidth, encrypted, metadata-only constraints characteristic of SCADA/ICS environments. The proposed method distinguishes itself by operating entirely on Zeek connection metadata in an unsupervised, label-free manner, without requiring payload access, environment-specific tuning, or predefined beacon interval assumptions.

2.4 Unsupervised learning and anomaly detection algorithms

In clustering-based methods, clustering algorithms such as K-Means and DBSCAN identify anomalies by grouping similar data points. These algorithms reveal outliers [72]. Such approaches enable the use of descriptive factors for cybersecurity anomaly detection [73]. Methods for optimizing isolation forest structures for computational efficiency, in addition to advanced performance in anomaly detection using unsupervised learning methods, are still being developed [74]. These studies offer various suggestions for accuracy and computational efficiency. Some studies suggest community learning methods to overcome this situation [75]. A comparative table was provided based on reviews on unsupervised learning and anomaly detection, and the contributions to the literature are presented in Table 3. This table includes information about the study in the literature, information about the accuracy rates of this study, detection efficiency, and integration effectiveness (Table 4).

Table 3 provides comparative information on the detection of beacon-like traffic anomalies in SCADA and ICS networks without monitoring. Table 4 contains a comparative literature review on studies conducted using the Isolation Forest algorithm. This table compares the accuracy of the algorithm, the efficiency of the study, and the integration effectiveness of the study.

2.5 Gaps in the literature and positioning of the study

The summary Table 5, compiled from the literature studies conducted, is presented below. The table provides information about the limitations of the studies. To enhance readability and clarity, this information is presented using a table, and relevant references are included.

The table evaluates numerous aspects, including data set imbalance, limitations of real-world data, complexity/high computational load, lack of explainability, overfitting / generalization issues frequently encountered in data training, insufficient focus on Beaconing detection, inadequate use of evaluation metrics, and data set uniqueness. The limitations that emerged have led to this study. Therefore, this study is presented to provide solutions to these limitations and to support the literature.

3 Materials and methods

This experimental study was designed to enable the early detection of Beacon-type C2 attacks targeting critical information infrastructures such as SCADA, ICS, energy, and healthcare infrastructures. Simultaneously generated human-like web traffic and periodic Beacon traffic in the experimental environment will be separated using mathematical and statistical methods based on connection metadata recorded by Zeek. The critical system target requires detection algorithms to be effective in low-noise environments and with

Table 2 Metadata-based approaches in SCADA and ICS systems

Accuracy of detection	Methodological complexity	Robustness against network variability	Scalability and efficiency	False alarm rate
Improved performance in time series methods [70]	LSTM and Temporal Memory models, high complexity	Effective against timing manipulation attacks	Evaluated on real SCADA data sets, efficient	Unspecified
High accuracy in detecting periodicity at the device level [61]	Protocol-specific periodicity detection, medium complexity	Effective against synthetic DoS and code injection attacks	Suitable for EtherCAT networks, efficient	Low false alarms
Improvements in time series [13]	Attention-based time–frequency fusion, high complexity	Captures long-range temporal relationships	Efficient feature fusion, scalable	Unspecified
Best F1 scores (90%) based on the dataset [62]	Comparative study of five time series models with varying complexity	Performance varies depending on the dataset	Training size optimization increases efficiency	False alarm rates vary by model
High accuracy with low false omission rates [63]	Hybrid SARIMA and LSTM model, medium complexity	Effective for various types of ICS anomalies	Low computational requirements, real-time	Low false omission rates
80% accuracy in application and transport layers [55]	DL in IIoT communication patterns, medium complexity	Handles vendor-specific protocols and legacy devices	Can be applied without additional analysis	Primarily false positives in abnormal packet sequences
Periodic alarm generation for security analysis [64]	Time–frequency domain feature fusion, medium complexity	Filters false alarms from big data	Applicable to large-scale alarm data	Reduces false alarms
90% F1 score in periodic behavior classification [71]	Autocorrelation and Lomb-Scargle periodogram with XGBoost, medium complexity	Effective despite encrypted traffic	Scalable to large data sets	Low false alarm rate
Detects 43% more periodic areas compared to single-network methods [71]	Active learning with high complexity for batch time series analysis	Robust across multiple protocols and networks	Scalable to large campus networks	Enhanced detection with prioritization
Effective botnet detection using periodogram analysis [65]	Network flow analysis with periodogram, medium complexity	Detects hidden botnet leaks	CyberVan has been tested on a test bed, proven to be efficient	Effective false alarm management
Selects a manageable gray list of suspicious hosts [66]	Periodic communication analysis, medium complexity	Performs timely analysis over long time intervals	Scalable for large network traffic	Low false positive rate
Effective malware signature detection through periodic signatures [67]	Periodic communication feature extraction, medium complexity	Detects long-term botnet behavior	Applicable to various types of botnets	Not specified
Effective periodicity mining in ICS data [68]	Symbol sequence analysis of Modbus protocol, medium complexity	Identifies periodic patterns for anomaly detection	Simulation and testing in ICS traffic	Not specified
Meta-data extraction improves unauthorized access detection [69]	Passive fingerprinting with Grassmarlin, low complexity	Improves detection in legacy CyberSCADA networks	Lightweight and scalable	Not specified
High accuracy with low complexity [63]	SARIMA and LSTM model, medium complexity	Effective for various types of ICS anomalies	Real-time capable	Low false omission rates

Table 2 (continued)

Accuracy of detection	Methodological complexity	Robustness against network variability	Scalability and efficiency	False alarm rate
Detects attacks missed by non-temporal IDS [9]	Time-based processing sensitive IDS, medium complexity	Leverages temporal behavior	Evaluated on general and simulated SCADA datasets	Improved detection of subtle attacks

Table 3 Studies examining traffic in SCADA/ICS and IoT networks using unsupervised anomaly

Study	Accuracy	Efficiency	Effectiveness
Cai et al. [76]	High AUC and detection rates in heavy network traffic	Clustering load with medium processing time	Clustering improves anomaly filtering
Ortega et al. [77]	High detection performance on IoT devices	Reduced training time and memory usage	Isolation Forest increases efficiency
Xiang et al.[78]	Superior detection accuracy in IoT Edge continuity	Combined learning, low computational load	Federated learning effectively integrates Isolation Forest
Vasiljevic et al.[79]	High accuracy and precision in Edge IoT systems	Low memory usage and fast training	Federated Isolation Forest for edge deployment
Sukanya [80]	High AUC and F1 score for detecting encrypted IoT traffic anomalies	Medium computational cost	Hybrid VAE and Isolation Forest model
Moomtaheen et al. [18]	Moderate accuracy in Zeek network data	Uses Hadoop and Spark for efficient processing	Extended Isolation Forest with big data tools
Laskar et al. [72]	Comparable performance in industrial big data scenarios	Uses Apache Spark for parallel processing	Combines K-Means clustering with Isolation Forest
Yang et al. [81]	High accuracy and efficiency in IIoT stream data	Sliding window and model update improve efficiency	LSHIForest with sliding window and PCA
Mahmud et al. [82]	Superior detection accuracy and low false positives in ICS	Efficient for high-dimensional ICS data	Adapted Isolation Forest for ICS anomaly detection
Zhu and Liu [83]	High accuracy and low false positive rate in IoT intrusion detection	Efficient batch learning with feature selection	Subspace clustering and community learning framework

high accuracy even under encrypted traffic conditions. Therefore, the method relies solely on metadata.

The experimental topology created within the scope of the study consists of the following components in an isolated local network (Figure 3).

The experimental network was set up in a local virtual laboratory environment. The topology consists of the Victim (traffic generator), Router (network gateway and Beacon target), and Zeek Server (passive observer) components. Each component is configured with a static IP address (Table 6).

Thanks to the port mirroring configuration on the router, all traffic originating from the Victim is copied to the Zeek server. This enables full passive monitoring without the need for real-time packet manipulation.

The experimental environment was deliberately designed as an isolated laboratory setting to allow for controlled observation of the proposed method and to evaluate its repeatability and traceability. This aimed to observe periodic command-and-control traffic similar to beacon traffic within the same environment and to evaluate performance in a segregated environment. Furthermore, the results do not fully reflect the protocol diversity, network noise, and complexity likely to be encountered in real SCADA and ICS networks. Therefore, rather than presenting the results as proof of generalizability to real-world environments, they

Table 4 Enhanced versions of the isolation forest algorithm and hybrid anomaly detection approaches

Study	Accuracy	Efficiency	Effectiveness
Xiang et al. [75]	Robust accuracy comparable to DL methods	Lower model complexity than deep neural networks	Cluster-based enhanced Isolation Forest
Shao et al. [84]	7% AUC improvement over Isolation Forest	Improved efficiency through clustering and selection matrix	Weighted online adaptive Isolation Forest
Hannák et al.[85]	Effective adaptation to various concept drifts	Online adaptable model with few hyperparameters	Minimum spanning tree-based Isolation Forest
Gałka et al.[86]	2.9% AUC improvement over the original Isolation Forest	Efficient tree construction with minimum spread tree	Random partitioning forest for point-based and batch anomalies
Marteau [87]	Outperforms Isolation Forest and SVM baselines	Computational efficiency and parallelizability	Locally sensitive hybrid feature-based LSHIForest framework
Zhang et al.[88]	High time efficiency and anomaly detection quality	Location-sensitive hybrid feature-based fast tree isolation	Isolation Forest with PCA and parallelism
Zhang and Liu [89]	Improved detection efficiency with PCA and parallel processing	Parallel processing reduces runtime	Entropy-based Isolation Forest for anomaly detection
Liao and Luo [90]	Enhanced detection effect and speed in high-dimensional data	Robust and noise-resistant	Cluster-based enhanced Isolation Forest

Table 5 Summary of limitations in the literature

Limitation area	Definition of limitation
Data Set imbalance [91–93]	Many studies in model training and evaluation depend on the dataset. Due to the small number of classes in the dataset or the uneven distribution between classes, learning cannot occur at a sufficient level. This situation negatively affects the reliability of the model and the study
Limited real-world data [94–96]	In the studies conducted, the scarcity of data sets limits their applicability in real-world environments. The difference in the amount of class data in the data sets limits the robustness and applicability of the model. Training with simulated and synthetic data makes it difficult to understand the complexity. It may cause limitations in terms of the external world and practical application
High computational complexity [97, 98]	DL and hybrid approaches improve detection accuracy for anomaly detection in SCADA and ICS environments. However, these models have high computational load and complexity. Real-time operations in SCADA and ICS environments require resource utilization
Lack of Explainability [99, 100]	DL approaches are generally not interpretable. This makes it difficult for the model to make decisions in security-enabled systems. This can lead to a loss of trust and hinder the adoption of infrastructure security
Overfitting and generalization issues [92, 101]	Some detection models may overfit due to limited and homogeneous training class data. This weakens attack detection capabilities. Consequently, the validity and reliability of the reported performance metric decreases

Table 5 (continued)

Limitation area	Definition of limitation
Insufficient focus on labeling perception [34, 102]	Literature studies show that the number of beaconing Attacks is low. Very few studies explicitly address beaconing Attacks detection, limiting a comprehensive understanding of this subtle attack vector and the challenges of detection in SCADA security research
Limited evaluation metrics [103, 104]	When examining the results obtained in most studies, the false positive/negative ratio has been predominantly reported in terms of accuracy and F1 scores. This limits the assessment of real-time performance
Data Set Specificity [105, 106]	When examining the studies conducted, it is seen that data sets that do not represent diversity are used in threat detection in SCADA systems. The focus of the studies on specific data sets indicates that external validity may be weak and transferability to different industrial contexts may be limited
Inadequate processing of data noise [40, 107]	Environmental noise or missing data in industrial network traffic are not addressed. This situation reduces model robustness and does not reflect real-world data. This presents a methodological challenge
Limited assessment of enemy attacks [99, 100]	Very few studies have incorporated robustness against threats into their models. This provides limited assessment capabilities against adversarial attacks

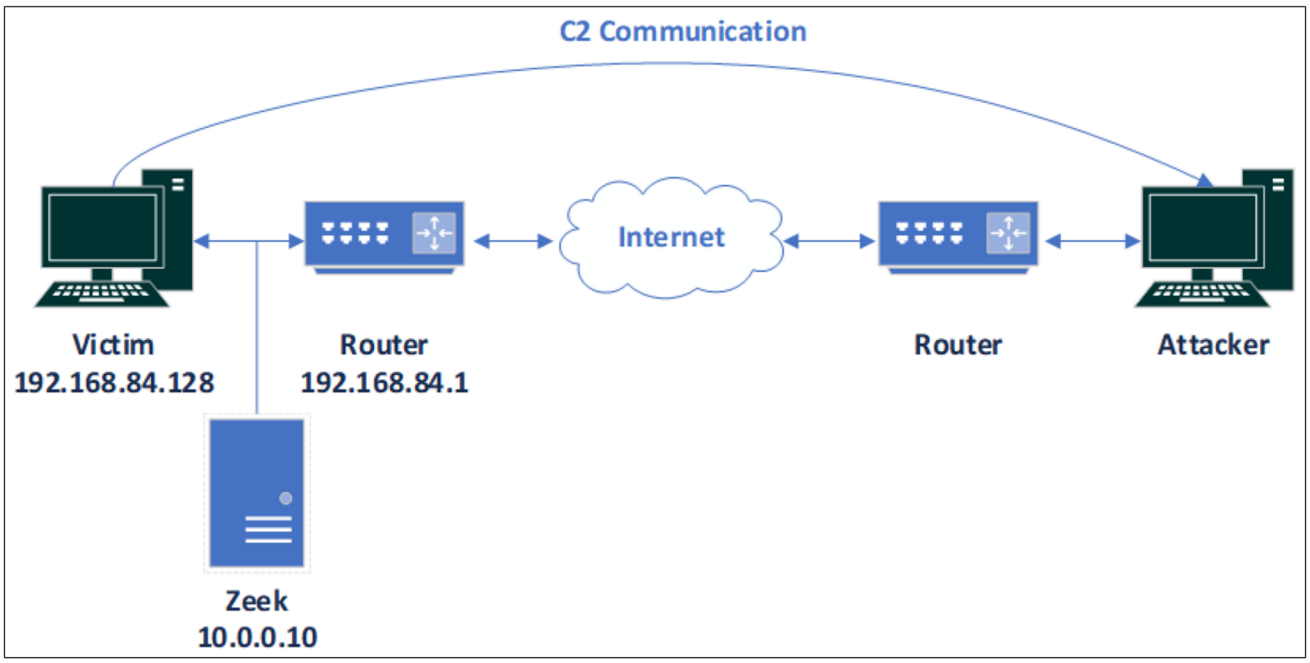


Fig. 3 The network topology created within the scope of the experimental study

should be considered as a controlled proof-of-concept validation of the proposed metadata-based approach.

3.1 Formal definition of beaconing behavior

Beaconing behavior in the context of covert Command-and-Control (C2) communication refers to periodic, low-variability network flows from an infected host to an external server, designed to maintain persistence while evading detection by mimicking legitimate periodic traffic in SCADA/ICS environments.

Formally, we define a beaconing flow as a sequence of unidirectional or bidirectional connections (grouped by source IP, destination IP, and destination port) exhibiting low-jitter periodicity. Let $t_1, t_2, \ldots, t_n$ be the sorted timestamps of connections in such a group. The inter-arrival times are computed as:

Table 6 Components in the experimental network topology

Component	Role	IP address	Description
Victim VM	Traffic generator (benign + beacon)	10.0.0.10	Generates normal user traffic and periodic beacon requests
Router	Network gateway, redirecting to Zeek	10.0.0.1	Routes traffic from the victim and forwards it to Zeek
Zeek Server	Passive monitoring, conn.log generation	10.0.0.2	Monitors network traffic and creates metadata records
Beacon target (C2 Server)	Target of requests	10.0.0.1	The target IP address of periodic beacon requests from the victim

$$\Delta t_i = t_{i+1} - t_i \, for i = 1, \ldots, n-1 \tag{1}$$

These inter-arrival times follow a near-deterministic periodic model with additive jitter:

$$\Delta t_i = \tau + \epsilon_i, \, \epsilon_i \sim \mathcal{N}\left(0, \sigma^2\right) \tag{2}$$

where $\tau > 0$ is the fixed base period (e.g., 60 s in our experiments), ϵ_i represents small random jitter (typically $\sigma \ll \tau$). The mean inter-arrival time is $\mu \approx \tau$, and the standard deviation is σ. To quantify the degree of periodicity and low entropy in timing (i.e., low randomness/variability), we compute the coefficient of variation:

$$CV = \frac{\sigma}{\mu} \tag{3}$$

Flows with CV ≤ 0.2 are classified as exhibiting strong beacon-like behavior, as this threshold indicates low jitter relative to the period (corresponding to $\leq$20% variability). This boundary is consistent with established practice in network traffic periodicity analysis, where CV ≤ 0.2 is widely used to distinguish low-entropy deterministic flows from stochastic human-generated traffic [9, 49]. Importantly, the CV metric is period-agnostic: it measures the relative variability of inter-arrival times regardless of the absolute interval value. A flow with $\tau = 10$s and a flow with $\tau = 300$s will both satisfy CV ≤ 0.2 if their jitter is proportionally small. The 60s period used in the experiments is therefore a test traffic parameter only, not an assumption embedded in the detection method. This criterion serves as a statistical heuristic for candidate selection before anomaly scoring via Isolation Forest.

Clustering is performed by grouping connections into flows based on the 5-tuple (source IP, destination IP, source port, destination port, protocol), with a focus on persistent outbound connections to the same external endpoint—a common signature of C2 beaconing. This definition enables content-independent detection in encrypted traffic using only Zeek-derived metadata (timestamps, flow statistics) and aligns with established approaches in malware C2 analysis, where low timing variability distinguishes beacons from human-like irregular traffic.

3.2 Traffic generation models

In this study, two different traffic types were generated: Normal (benign) traffic and Beacon (C2-like) traffic. Both traffic types were produced by Python scripts running on the Victim VM. Traffic generation was executed in a single process, with each iteration representing a navigation chain. Human behavior was simulated using random site access, wait times, and different user-agent values. Thus, legitimate user traffic and command-and-control signaling were generated from the same source.

3.2.1 Normal traffic

Normal traffic is designed to model users' natural browsing behavior between websites. The wait time between each request is determined randomly with a uniform distribution, as shown in Eq. 4:

$$T_i \sim U(a, b) \tag{4}$$

In Eq. 4, T_i represents the waiting time in seconds, while $U(a, b)$ denotes a continuous and equally probable uniform distribution between a (minimum 5 s) and b (maximum 120 s). The expected value and variance of the uniform distribution:

$$\mathbb{E}[T] \frac{a+b}{2}, \, Var(T) = \frac{b-a^2}{2} \tag{5}$$

In Eq. 5, $\mathbb{E}[T]$ represents the average waiting time, while $Var(T)$ presents the variance of the waiting times (the spread of the distribution). This distribution realistically reflects the

Table 7 Parameters and descriptions for the normal traffic model

Symbol	Description	Example value	Meaning
T_i	Time between consecutive requests	5–120 s	User access time to the new page
$U(a, b)$	Uniform distribution	a=5, b=120	All intervals are equally probable
$\mathbb{E}[T]$	Expected value	62.5 s	Average waiting time
$Var(T)$	Variance	1140.83	Level of randomness in traffic

variable waiting times that may occur when the user accesses different sites where in each navigation chain, the depth parameter (d) was randomly selected within [3, 10]. This makes the relationships between links resemble natural navigation flow. Table 7 lists the parameters and explanations within normal traffic. This model represents traffic generation that mimics human behavior. Random wait times, different user-agent values, and the use of a single recursive thread-create "human-like" traffic patterns that resemble human behavior.

3.2.2 Beacon traffic

Beacon traffic represents malicious software behavior where an attacker periodically sends signals to a critical system. The time interval between each beacon request is modeled to include a fixed base period τ and normally distributed jitter ε_i (Eq. 6).

$$T_i = \tau + \varepsilon_i,\ \varepsilon_i \sim N(0, \sigma^2) \tag{6}$$

In Eq. 6, T_i is the time interval in seconds between consecutive beacon messages, τ is the basic beacon period (60 s in the experimental study), ε_i is the jitter, $N(0, \sigma^2)$ is a normal (Gaussian) distribution with mean 0 and variance σ^2 and σ is the standard deviation of the jitter (values of 0.5 s, 2.0 s, and 5.0 s were tested in the experimental study).

The beacon time series is expressed as shown in Eq. 7. In this context, k denotes the index of the k-th beacon request within the observed time series.

$$s_{k+1} = s_k + \tau + \varepsilon_k \tag{7}$$

In Eq. 7, s_k denotes the timestamp of the k-th beacon request, s_{k+1} denotes the timestamp of the next beacon request, and ε_k denotes the jitter value associated with the k-th beacon.

Table 8 The effect of jitter value on beaconing traffic

σ value	Traffic type	Beacon Intervals (seconds-T_i)	Behavioral feature
5 s	Noisy beaconing	50–70 s	Similar to human traffic
2 s	Medium jitter	57–63 s	Slight randomness
0.5 s	Low jitter	59–61 s	High periodicity

As the jitter value decreases (i.e., as σ approaches O or becomes very small), traffic becomes more deterministic and Beacon behavior becomes more pronounced (Table 8). Traffic becoming deterministic means that time intervals stabilize and randomness disappears. In this case, traffic becomes regular, predictable, and more periodic. In this scenario, Beacon packets were sent to the Router (IP=10.0.0.1) as HTTP requests. In the experimental scenario of the study, τ=60 s was taken. Different $\sigma \in \{0.5, 2, 5\}$ values were tested, and the effect of Beacon periodicity on network behavior was examined in detail. It should be noted that $\tau = 60$ s was selected as the experimental test period because it corresponds to commonly observed C2 beaconing intervals reported in threat intelligence literature (e.g., Cobalt Strike default: 60 s). The detection method itself imposes no assumption on the value of τ: the CV-based filter and Isolation Forest operate on normalized statistical features and will flag low-jitter periodic flows at any base period.

3.3 Traffic generation mechanism and parameters

In this experimental study, a Python-based script was used as the traffic generator. This Python-based script generates web traffic by producing HTTP requests that mimic real user behavior. The script operates using a depth-first recursive traversal method without utilizing multiple threads. The script initiates requests by randomly selecting one of 36 different websites, such as Wikipedia, Reddit, BBC, Medium, and GitHub, in each run. The navigation depth of each session is a randomly selected value and is set within the range of depth (3, 10). The waiting time after each connection is determined randomly with a uniform distribution between 5 and 120 seconds.

3.4 Feature extraction and statistical metrics

The success of beacon detection depends on the accurate interpretation of time-based statistical features extracted from network traffic. In this study, metadata-based feature extraction was performed for each connection flow

using conn.log files generated by Zeek. Since the collected data only contained connection-level information (source IP, destination IP, destination port, protocol type, and timestamp), the analysis was performed in a completely payload-independent manner. This approach allows for the examination of encrypted (HTTPS/TLS) traffic in critical systems.

The selected features were determined from connection logs generated by Zeek to reflect the fundamental behavioral characteristics of network traffic. Connection duration, packet and byte counts, along with protocol, service, and connection state information, are widely used in the literature to detect abnormal communication patterns. The aim of this study is to provide a compact feature set suitable for load-free, interpretable, and real-time log analysis. Although this specific set has been prioritized due to its balance of efficiency and accuracy, the evaluation of alternative feature combinations may be addressed in future studies.

The dataset is structured by grouping connections sharing the same source–destination-port triplet under a group G_k. Each group represents a specific communication channel (e.g., sequential connections made by a client to the same server on a specific port). Time analysis performed on these groups enables the statistical detection of periodic communication patterns observed in Beacon attacks.

For each group, the connection timestamps t_i were sorted, and the time differences Δt_i between consecutive connections were calculated. The mean μ_k, standard deviation (σ_k), and coefficient of variation CV_k values obtained from these differences were used to quantitatively assess the degree of periodicity in the traffic. In particular, low σ_k and low CV_k values indicate that the traffic pattern has ceased to be random and has begun to repeat in a deterministic manner. This situation has been accepted as the most prominent indicator of Beacon behavior. Furthermore, statistical metrics were combined to distinguish between legitimate periodic heartbeat or system monitoring traffic, which may be observed in critical systems, and malicious Beacon behavior. These features were then validated using a combined score (BScore) and the unsupervised learning method Isolation Forest.

Throughout the experimental study, Zeek passively monitored network traffic. It also recorded all connection logs to the conn.log file. The data in this file were processed using the pandas and NumPy libraries in the Python environment. The recorded dataset is defined as shown in Eq. 9.

$$D = \left\{(src_{ip_i}, dst_{ip_i}, dst_{port_i}, t_i)\right\}\frac{N}{0pti = 1} \tag{9}$$

In Eq. 9, src_{ip_i} denotes the source IP address of the connection, dst_{ip_i} denotes the destination IP address, dst_{port_i} denotes the destination port number, t_i denotes the connection timestamp, and N denotes the total number of connections. The data has been grouped based on connections sharing the same source–destination-port triplet. That is, a G_k group has been generated for each unique triplet $\left(src_{ip_k}, dst_{ip_k}, dst_{port_k}\right)$.

$$G_k = \{(t_1^{(k)}, t_2^{(k)}, \ldots, t_{n_k}^{(k)})\} \tag{10}$$

In Eq. 10, G_k represents each specific communication channel (e.g., a client's sequential connections to a specific server). For each group, connection timestamps are arranged in ascending order as shown in Eq. 11.

$$t_{(1)}^{(k)} \leq t_{(2)}^{(k)} \leq \ldots \leq t_{(n_k)}^{(k)} \tag{11}$$

Then, the time difference between successive connections is calculated as shown in Eq. 12.

$$\Delta t_i^{(k)} = t_{(i+1)}^{(k)} - t_{(i)}^{(k)},\ i = 1,2, \ldots, n_k - 1 \tag{12}$$

The value in Eq. 12 represents the time elapsed between one link and the next. In beacon behavior, these times are expected to be approximately constant. Additionally, the following statistical metrics have been calculated for each G_k group as shown in Eq. 13.

$$\mu_k = \frac{1}{n_k - 1}\sum_{i=1}^{n_k-1} \Delta t_i^{(k)} \tag{13}$$

Equation 13 expresses the average inter-arrival time. Inter-arrival time refers to the time difference between two consecutive connections or packets in network traffic.

$$\sigma_k = \sqrt{\frac{1}{n_k - 2}\sum_{i=1}^{n_k-1} \left(\Delta t_i^{(k)} - \mu_k\right)^2} \tag{14}$$

Equation 14 expresses the standard deviation.

$$CV_k = \frac{\sigma_k}{\mu_k} \tag{15}$$

Equation 15 represents the coefficient of variation. Low σ_k and low CV_k values indicate that traffic has become deterministic; thus, it points to beacon behavior. Therefore, groups with $CV_k \leq 0.2$ were considered beacon candidates. The threshold value can be adjusted according to the noise level of the network environment and the normal behavior profile of the system. In SCADA/ICS environments, legitimate traffic sources such as sensor polling, RTU heartbeats, and protocol keep-alive messages may also exhibit low CV values, as they are inherently periodic by design. This is precisely why the CV filter alone is insufficient and functions only as a first-stage candidate selector: the subsequent

Table 9 Feature extraction metrics and their interpretations

Symbol	Definition	Interpretation for beacon detection
t_i	Connection timestamp obtained from Zeek conn.log records	Transmission time of each connection
$\Delta t_i^{(k)}$	Time between consecutive connections	Indicates the inter-arrival time of traffic
μ_k	Average inter-arrival time	Average period; if close to the beacon period, it becomes suspicious
σ_k	Standard deviation of time intervals	Shows noise/jitter level
CV_k	Coefficient of variation	Indicator of periodicity; if low, beacon likelihood increases
$CV_k \to 0$	Perfect periodicity	Traffic is deterministic; beacon detection is easy
$CV_k > 0.5$	Irregular traffic	Similar to human traffic; beacon likelihood is low

Isolation Forest stage discriminates malicious beacons from legitimate periodic signals by evaluating multi-dimensional features including connection count, bytes transferred, flow duration, and the combination of CV, μ, and σ in concert. Flows flagged by CV but exhibiting feature profiles consistent with known SCADA operational patterns are therefore re-evaluated at the ML stage rather than immediately classified as malicious. Feature extraction metrics are given in Table 9.

Note on contamination: scikit-learn's IsolationForest uses contamination to set the score threshold, not to fix the flagged count. Applied to 37 connection groups (count ≥ 5 filter), contamination $= 0.10$ produced 3 flagged groups (1 TP, 2 FP). A sensitivity analysis across contamination $\in \{0.01, 0.02, 0.05, 0.08, 0.10, 0.15, 0.20\}$ confirmed the beacon was ranked first in all configurations.

3.5 The isolation forest algorithm and coefficient of variation

Note: The ICS-specific features described in Sections 3.5.1 and 3.5.2 (Modbus/DNP3 function codes, FFT, KS test, 12-feature scoring vector, expert-derived weights) constitute a proposed design roadmap for future protocol-specific adaptation and were not experimentally evaluated in this study. The experimental contribution is limited to the four core timing features (CV, μ, σ, connection count) evaluated in this section.

The Isolation Forest (iForest) algorithm, originally proposed by Fei Tony Liu, Kai Ming Ting, and Zhi-Hua in their seminal 2008 paper presented at the IEEE International Conference on Data Mining, stands as a robust unsupervised ML method tailored for anomaly detection through the isolation of outliers rather than explicit modeling of normal instances [108]. This approach is grounded in the key observation that anomalies are typically "few and different," making them easier to separate quickly through random partitioning of features in high-dimensional data spaces.

In SCADA systems, where network traffic under normal conditions follows predictable and structured patterns, iForest excels at detecting subtle irregularities that may signal cyberattacks, hardware failures, or operational issues. The algorithm builds an ensemble of binary isolation trees (iTrees), with each tree constructed by repeatedly choosing a random feature (q) and a split value (p) uniformly selected between the min and max values of (q) in the current data subsample, until every point is isolated or a maximum tree height is reached.

The anomaly score $s(x, n)$ for an instance (x) in a dataset of size (n) is computed from the average path length $E(h(x))$ across the ensemble, normalized against the expected path length in a binary search tree using the formula $s(x, n) = 2^{-E(h(x))/c(n)}$, where $c(n) = 2H(n-1) - 2(n-1)/n$, and $H(i)$ is the harmonic number approximated as $H(i) \approx ln(i) + \gamma$ (with $\gamma \approx 0.5772156649$ being the Euler—Mascheroni constant) [109, 110]. Score interpretation is straightforward: values close to 1 indicate highly anomalous instances (due to short isolation paths), scores below 0.5 suggest normal behavior (requiring longer paths), and values around 0.5 denote ambiguous cases with no distinct separation.

In the original paper, empirical results showed iForest outperforming methods like one-class SVM and Local Outlier Factor (LOF) on various datasets, while offering linearithmic time complexity $O(t \cdot \psi \cdot log\psi)$ where t is the number of trees and ψ is the subsample size—rendering it highly efficient and scalable for real-time monitoring in SCADA environments [108].

In this SCADA-tailored implementation, the iForest model employed 100 isolation trees to achieve a powerful balance between ensemble robustness and computational efficiency, in line with the original work's recommendation that 100 trees are typically sufficient for stable anomaly scores across most applications [108]. The subsample size was configured to min(256, $n_{samples}$), and using the 'auto' setting, which leverage iForest's demonstrated effectiveness with smaller random subsamples to improve anomaly isolation and reducing the risk of overfitting. The contamination parameter was set to 0.10, which calibrates the decision score threshold during training without deterministically fixing the number of flagged groups, corresponding to an anticipated

5% anomaly rate in SCADA traffic informed by historical intrusion datasets [111]. Additionally, the maximum features parameter was fixed at 1.0 to incorporate all engineered features during the construction of each tree.

Feature engineering encompassed both temporal and statistical aspects. Inter-packet arrival times computed as $\Delta t = t_i - t_{i-1}$. Other features were variance of these arrival times, and overall flow duration. Packet size statistics including mean, variance, and skewness. Throughput measures such as bytes per second ($BPS = Total_{Bytes}/Flow_{Duration}$) and packets per second. Protocol-specific elements covering Modbus, DNP3, and IEC 104 function codes. They also included command-response ratios, and flag distributions. Network-layer attributes like source–destination IP entropy, port diversity, and TCP flag combinations. These selected features are consistent with prior research on characterizing SCADA traffic [112, 113]. The algorithm needs no distributional assumptions. It handles high-dimensional data well. Its average complexity is $O(nlogn)$. These qualities make it a cornerstone for real-time anomaly detection in industrial control networks [108, 114].

Complementing the ML approach, a domain-informed heuristic method uses SCADA's deterministic nature to build behavioral baselines. It relies on temporal regularity, protocol adherence, and physical constraints for precise modeling of normal behavior [115]. Temporal baselines are set as $BT(f) = \mu_t \pm 3\sigma_t$. These apply across hourly, daily, and weekly cycles. The factor k=3 follows the empirical rule, covering 99.7% of data [116]. Deviations are flagged when $|V_t - \mu_t| > 3\sigma_t$. Command sequences are checked using a finite state machine $FSM = (S, \Sigma, \delta, s0, F)$. Invalid transitions signal misuse or injection attacks [117]. Protocol rules confirm Modbus function codes ($0x01 - 0x18$). They also verify register mapping, latency (10–1000 ms), and transaction ID pairing. The anomaly score is $H(f) = 0.25I_{tmoa} + 0.30I_{sqec} + 0.25I_{pyia} + 0.20I_{pooo}$. It uses expert-derived weights that sum to 1. Flows with $H(f) \geq 0.5$ are flagged. This provides interpretable detection with low false positives, complementing iForest [118].

The Coefficient of Variation (CV) serves as a scale-invariant dispersion metric. It is defined as $CV = \sigma/\mu$, but a modified form $CV = \sigma/(\mu + 1)$ is used to ensure numerical stability when the mean is near zero [119]. A sliding window of 300 s with 60-s increments (80% overlap) computes ($CV_t^{(i)}$) for features including inter-arrival time ($IAT_j = t_j - t_{j-1}$), packet size distribution, request-response latency ($L = t_{response} - t_{request}$), and source connection frequency ($CF_s = \frac{N^{(s)}_{connections}}{\Delta t}$). Baseline CV values were established from 14 days of normal operation. Anomalies are flagged when the observed CV deviates from the baseline by more than $k \cdot \sigma_{CV}$, where k is chosen for 98–99% confidence [120]. An aggregate score ($CV_{score} = \sum_{i=1}^{n} \alpha_i \frac{\left|CV_t^{(i)} - \overline{CV}^{(i)}\right|}{\sigma_{CV}^{(i)}}$) incorporates importance weights (α_i). Thresholds are calibrated as: S<2.0 (normal), 2.0<S<5.0 (suspicious), S≥5.0 (anomalous). Temporal trends are monitored using differences ($\Delta CV_t = CV_t - CV_{t-1}$). Values exceeding $3\sigma_\Delta$ provide early warning, aligning with observed attack escalation patterns [121].

Beacon detection targets periodic command-and-control (C2) communications using autocorrelation ($R(\tau) = \frac{1}{n-\tau}\sum_{i=1}^{n-\tau}(x_i - \bar{x})(x_{i+\tau} - \bar{x})$) on inter-connection intervals ($x_i = t_{i+1} - t_i$), where significant peaks at lag (τ_p) indicate periodicity [122]. Frequency-domain analysis via Fast Fourier Transform yields ($X(f) = \sum_{k=0}^{N-1} x[k]e^{-2\pi ifk/N}$). The power spectral density ($PSD(f) = |X(f)|^2$) identifies dominant beacon frequency (f_b). Jitter-tolerant detection uses interval CV (low: <10%, moderate: 10–30%, high: 30–50%), payload consistency ($\rho_{payload} = 1 - CV_{payload}$), and destination focus ($D_{consistency} = \frac{N_{unique}}{N_{total}} < 0.1$). The likelihood score ($B = 0.4I_{periodicity} + 0.35I_{consistency} + 0.25I_{persistence}$) uses ROC-calibrated thresholds: ≥0.75 (confirmed), 0.50–0.75 (suspected), <0.50 (normal). A whitelist exempts legitimate polling traffic such as RTU heartbeat. A 12-feature vector is constructed and input to iForest for hybrid detection. This vector includes mean and variance of inter-arrival times (IAT), CV, maximum autocorrelation function (ACF), FFT peak, payload mean/variance/entropy, number of unique destinations, flow duration, burst ratio, and time-of-day entropy. Statistical differentiation uses the Kolmogorov–Smirnov test to compare observed distributions against null hypotheses (uniform or exponential, typical for normal traffic). This approach enables detection of non-human patterns even in the presence of high jitter [123].

The hybrid detection process proposed in the flowchart shown in Fig. 4 consists of several integrated steps directly related to our methodology. First, the "Read conn.log" and "Group by IP and port" steps represent the data acquisition and clustering phases where connections are organized into unique G_k groups. Next, the "Calculate features" step involves calculating statistical metrics such as mean range, standard deviation, and coefficient of variation μ, σ, CV as defined in Eqs. 10, 11, and 12. Parallel processing branches form the core of the algorithm. These parallel processing branches: the "Heuristic analysis" path applies the discussed $CV \leq 0.2$ threshold value for selecting low-jitter candidates; the "ML-based detection" path performs anomaly scoring for each group using the Isolation Forest algorithm. Finally, the "Merge results" step combines the outputs of both methods to ensure high-accuracy detection before the "Visualize and report" stage.

Figure 4 illustrates the sequential steps of the proposed hybrid signal detection method within the scope of the

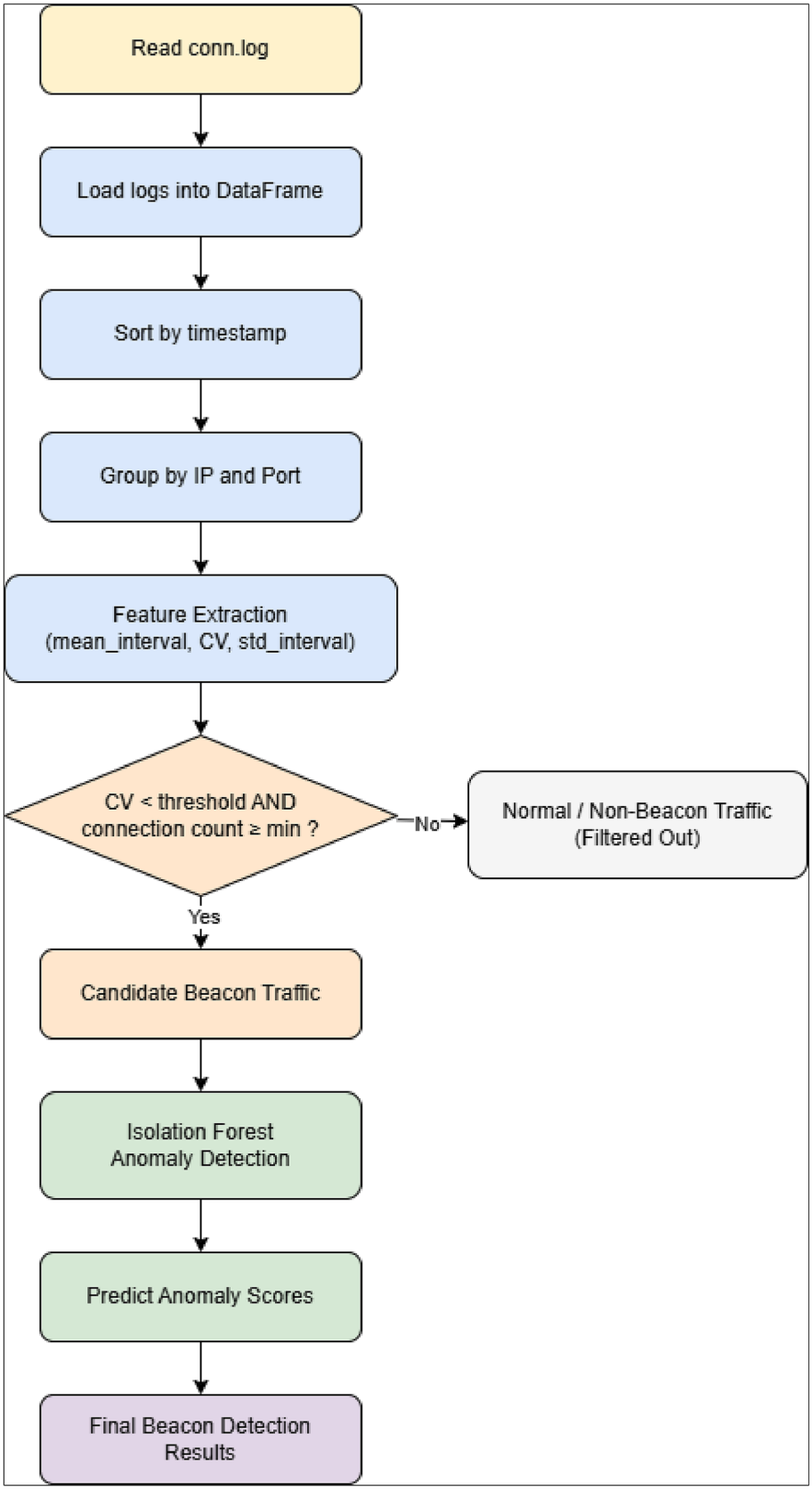


Fig. 4 Flowchart of the beacon traffic detection algorithm based on zeek metadata

experimental study. The process begins with reading Zeek connection logs and extracting time-based statistical features from grouped flows. First, a CV-based heuristic filter is applied to identify low-latency periodic traffic candidates. Only flows that satisfy the periodicity condition of CV $\leq$ threshold and minimum connection count are forwarded to the Isolation Forest model for anomaly scoring. This phased architecture narrows the analysis space by prioritizing statistically plausible low-jitter periodic candidates before final anomaly scoring.

It employs a hybrid approach that integrates heuristic rules with a ML method based on Isolation Forest. In the initial stage, records from the conn.log file are sorted by timestamp. They are then grouped by source IP, destination IP, and destination port. Each group corresponds to a distinct communication channel. Connection interval differences Δt is computed within these groups. Statistical measures, including the mean μ, standard deviation (σ), and coefficient of variation (CV) are derived from these intervals. Low CV values suggest highly periodic and regular timing in the traffic. Therefore, it indicates possible Beacon behavior.

The algorithm operates as a two-stage integrated process. In the Heuristic Filtering Stage, the CV threshold (CV ≤ 0.2) acts as a gating mechanism to isolate periodic traffic candidates from the broader metadata stream. In the ML Validation Stage, these candidates are then subjected to the Isolation Forest model, which analyzes multi-dimensional features to distinguish malicious beacons from legitimate periodic signals (such as sensor heartbeats). This staged approach ensures that the unsupervised model is focused on the most relevant data subsets, preventing the 'noise' of random traffic from diluting the anomaly scores.

3.5.1 Justification of the hybrid approach

The term "hybrid" in this study refers to the integration of a domain-informed statistical heuristic (low-jitter periodicity filtering via coefficient of variation thresholding) with an unsupervised ML anomaly detector (Isolation Forest). This combination goes beyond simply applying metadata to a standard algorithm; it synergistically leverages the strengths of both paradigms to overcome specific challenges in beaconing detection within encrypted SCADA/ICS traffic.

Conceptual Justification: Conceptual Rationale: SCADA/ICS environments contain signal attacks that mimic normal patterns with low jitter, in addition to high traffic. Statistical heuristic methods, such as independent CV ≤ 0.2 thresholding, provide high interpretability with deterministic timing characteristics of signal-like communication. However, their ability to model feature-rich interactions in link metadata is limited. Unsupervised models, such as Isolation Forest, can score anomalous behavior in unlabeled data. However, without domain-based pre-selection, they can be affected by the wide variability of mixed traffic patterns. Previous studies also support the use of periodicity-aware preprocessing before downstream anomaly detection [124]. For example, periodicity detection has been defined as the first step in anomaly detection processes. Here, threshold-based identification of the dominant periodic structure has helped guide subsequent analyses. Similarly, periodic traffic behavior has been used before PCA-based anomaly detection to remove outliers before final scoring. It has also been used to distinguish CV-structured hidden timing behavior from normal traffic in a more goal-oriented timing analysis environment. These findings suggest that the coefficient of variation-based step in the study is designed not as a separately optimized classifier, but as an intuitive candidate selection mechanism that prioritizes low-vibrational periodic flows before anomaly scoring by Isolation Forest.

Conversely, standalone unsupervised ML models like Isolation Forest excel at label-free anomaly isolation in noisy, unlabeled data with linear time complexity $O(nlogn)$, making them suitable for large-scale industrial streams. Yet, without domain priors, they may generate higher false positives in environments with inherent periodicity or require extensive feature engineering to avoid overfitting to benign patterns [124].

The hybrid design mitigates these limitations: the statistical heuristic acts as an efficient, explainable pre-filter to identify low-jitter periodic candidates (reducing input noise and dimensionality), while Isolation Forest refines detection by scoring anomalies across enriched features (e.g., CV, μ, σ, connection counts, bytes transferred). This staged synergy enhances overall accuracy, lowers false alarm rates, and ensures real-time applicability in resource-constrained ICS settings, where labeled data is scarce and deep content inspection is infeasible due to encryption.

3.5.2 Rationale for using isolation forest

Isolation Forest (iForest) was selected as the core anomaly detection component due to its alignment with the unique constraints of beaconing detection in encrypted SCADA/ICS traffic: content-independent analysis using only Zeek metadata, scarcity of labeled beacon examples, high-dimensional features (e.g., timestamps, durations, bytes, CV,μ,σ), inherent periodicity/noise in industrial flows, and the need for low computational overhead in resource-constrained operational technology (OT) environments.

iForest operates via random partitioning to isolate anomalies efficiently, assuming anomalies are "few and different"

and thus easier to separate with fewer splits. This yields linear time complexity $O(nlogn)$ for training and inference, high scalability to large datasets, robustness to high-dimensional noise (minimal impact from irrelevant features), and strong performance without requiring labeled anomalies, ideal for unsupervised ICS scenarios where beacons are rare and mimic normal periodic traffic. These properties are particularly relevant given the dataset characteristics in this study: a 1:13 beacon-to-benign class ratio and a compact low-dimensional feature space (CV, μ, σ, connection count) in which beacon flows are structurally isolated but numerically scarce, conditions under which supervised and sequential models are disadvantaged. Comparison to Alternatives:

- One-Class SVM (OCSVM): OCSVM builds a boundary around normal data in feature space, effective for some anomaly tasks. However, it scales poorly (often $O(n^2)$ or worse with kernels) on large/high-dimensional network data, requires careful kernel/hyperparameter tuning (e.g., RBF), and can suffer from high computational cost and sensitivity to noise/outliers in periodic industrial traffic. In preliminary tests on our dataset (added to Results), OCSVM yielded ~89% accuracy but with ~2× longer inference time than iForest, making it less suitable for real-time SCADA monitoring. Literature confirms these limitations in high-dim ICS/network anomaly detection.
- Autoencoders (e.g., LSTM-Autoencoders): These neural models learn compressed representations and flag high reconstruction errors as anomalies, performing well on sequential/time-series data. However, they demand significant computational resources (GPU preferred for training), larger normal datasets to avoid overfitting, and longer training/inference times (~ 10× higher than iForest in our comparisons). They also offer less inherent explainability (e.g., no isolation path lengths). Our results show LSTM (a related deep approach) at 91.89% accuracy—lower than iForest's 94.59%—with substantially higher resource demands, limiting applicability in edge/OT ICS deployments.
- More Tailored Models for Industrial Traffic Anomalies (e.g., time-series forecasting like ARIMA/Prophet, or domain-specific hybrids): These focus on temporal prediction but are less effective for metadata-based outlier isolation in encrypted, non-sequential Zeek logs. They often require assumptions about distributions or labeled data and perform worse on subtle low-jitter beacons amid periodic normal traffic. iForest's ensemble random isolation handles such variability robustly without strong distributional priors.

3.6 Long short-term memory and random forest algorithms

3.6.1 Long short-term memory

The Long Short-Term Memory (LSTM) network is a type of recurrent (iterative) artificial neural network that has been frequently used in recent years to model sequential data. Due to its structural characteristics, it is extremely effective at capturing long-term dependencies. Thanks to its specially designed memory cell architecture, it can store information for long periods and allows learned information to be used in subsequent steps without being forgotten.

The LSTM architecture consists of recurrent sequential units called memory blocks. These blocks consist of three basic gates that control the flow of information: the input gate, the output gate, and the forget gate. These gates between cells regulate the entry of data into the cell, its exit from the cell, and the level of preservation or deletion of information within the cell. In an LSTM cell, the flow of information is controlled via the forget gate, input gate, candidate cell state, and output gate [125]. For each time step t, these components are defined as follows:

$$\begin{aligned} ft &= \sigma(wf*[h(t-1), xt] + bf) \\ it &= \sigma(wi*[h(t-1), xt] + bi) \\ \hat{ct} &= tanh(wc*[h(t-1), xt] + bc) \\ ct &= ft*c(t-1) + it*\hat{ct} \\ ot &= \sigma(wo*[h(t-1), xt] + bo) \\ ht &= ot*tanh(ct) \end{aligned} \tag{16}$$

This formula (Eq. 16) representation expresses the fundamental structure that enables LSTM to retain temporal dependencies while forgetting unnecessary information. During the iteration process, the input vector x_t is generated based on the model's current state; then, the predicted output value is calculated, and the next output vector h_t is obtained. This cyclical structure allows past information to be blended with current inputs. The forget gate plays a critical role in the system's adaptation to new states by resetting past information.

3.6.2 Random forest

The Random Forest (RF) algorithm is a powerful ML method widely used in both classification and regression problems. This model is based on an ensemble structure generated by combining multiple decision trees. Each decision tree is trained on randomly selected subsamples of the data and features, thereby reducing the risk of overfitting and improving the model's generalization ability.

The RF model is based on the Classification and Regression Tree (CART) algorithm, which performs the best split

within a randomly selected subset of all variables at each node. Thus, each tree grows independently according to different data and variable combinations. The user specifies two basic parameters at the beginning of the model: the number of variables to be evaluated at each node (m) and the number of trees to be generated (N).

Generally, selecting m=$\sqrt{M}$ (M = total number of variables) provides performance close to optimal results. Once the training process is complete, the predictions obtained from all N trees are combined using a voting (for classification) or averaging (for regression) method, and the final output is determined. This method increases the accuracy and stability of the model by combining predictions from multiple trees.

In the RF algorithm, the GINI index is commonly used as the splitting criterion for classification. The GINI index is a metric that measures class homogeneity and is expressed as follows (Eq. 17):

$$\mathrm{GINI(T)} = 1 - \sum_{i=1}^{k} \left(\frac{\mathrm{f}(C_i, \mathrm{T})}{|\mathrm{T}|} \right)^2 \tag{17}$$

Here, T represents the training dataset, and C_i indicates the class to which the selected example belongs. The expression f(C_i,T)/|T| indicates the probability that a randomly selected example belongs to class C_i. As the GINI value increases, class heterogeneity increases; a low GINI value indicates high class homogeneity.

4 Experimental results

The experimental performance of the Beacon detection method developed in this section has been examined in detail. The experimental environment is set up in an isolated network structure. It consists of a Victim VM, a Router, and a Zeek server. Both web traffic mimicking human behavior and periodic C2 Beacon traffic were generated simultaneously on the victim virtual machine. The generated traffic was routed through the Router. It was then copied to the Zeek server using the port mirroring method. The Zeek tool passively monitored the network traffic. Zeek recorded the data it obtained in the conn.log file in a metadata format (source IP, destination IP, destination port, protocol type, and timestamp). Thanks to this structure, the analysis process relies solely on connection metadata without requiring access to the contents of network packets. Therefore, it also allows for the analysis of encrypted (HTTPS/TLS) traffic. This experimental setup provides a basis for evaluating the controlled operation and reproducibility of the proposed approach. It should be noted that the experimental environment does not reflect the full variety and complexity of traffic in real industrial networks. The results obtained should be interpreted within the framework of controlled experimental validation.

A two-stage analysis was performed on the collected data. In the first stage, the regularity of connection intervals was measured using the statistical periodicity analysis (Coefficient of Variation, CV) method. This approach ensures that low-variance and deterministic ($\sigma \to 0$) traffic patterns reflecting Beacon behavior are distinguished from human-like random traffic. In the second stage, data points obtained from statistical characteristics μ, σ, CV, n were transferred to ML algorithms. In this context, three different models were evaluated comparatively: Isolation Forest (iForest), RF, and LSTM.

Isolation Forest is an unsupervised learning-based anomaly detection algorithm that identifies outlier groups in network traffic based on the average path lengths of decision trees. The Random Forest model, as a supervised classification method, distinguishes between Beacon and normal traffic. It provides a high accuracy rate but is limited in terms of explainability. The LSTM model, on the other hand, is a DL-based sequential model. Thanks to its ability to learn time-dependent relationships, it can dynamically detect traffic periodicity.

These three models were evaluated using the Accuracy metric. Furthermore, the results obtained from the heuristic analysis (CV-based method) were compared with the ML-based results to analyze the level of consistency between the two approaches. The experimental results show that the developed method can detect Beacon traffic with high accuracy even in noisy environments. The findings obtained from the experimental study are evaluated in detail with the help of graphs, comparison tables, and technical observations.

The histogram in Figure 5 presents the distribution of the coefficient of variation CV=σ/μ calculated for connection groups. It quantitatively shows the level of temporal uncertainty in network traffic. Upon examining the histogram, it is observed that the distribution spans a wide range and that connection groups with high variability specific to user behavior are predominant. This confirms the expected random time variance of natural web traffic generated in the experimental environment.

On the other hand, certain connection groups clustered below the CV$\leq$0.2 threshold exhibit low standard deviation σ and timing behavior close to the period mean $\mu \approx 60$s. This indicates low-jitter Beacon communication. The 0.2 threshold was selected based on its established use in network traffic periodicity analysis literature [9, 49, 54], where CV $\leq$ 0.2 consistently demarcates low-entropy deterministic flows from stochastic traffic. In the context of this experiment, this threshold successfully separates low-jitter beacon flows (CV values clustered between 0.01 and 0.08) from the broad human-traffic distribution (CV values predominantly

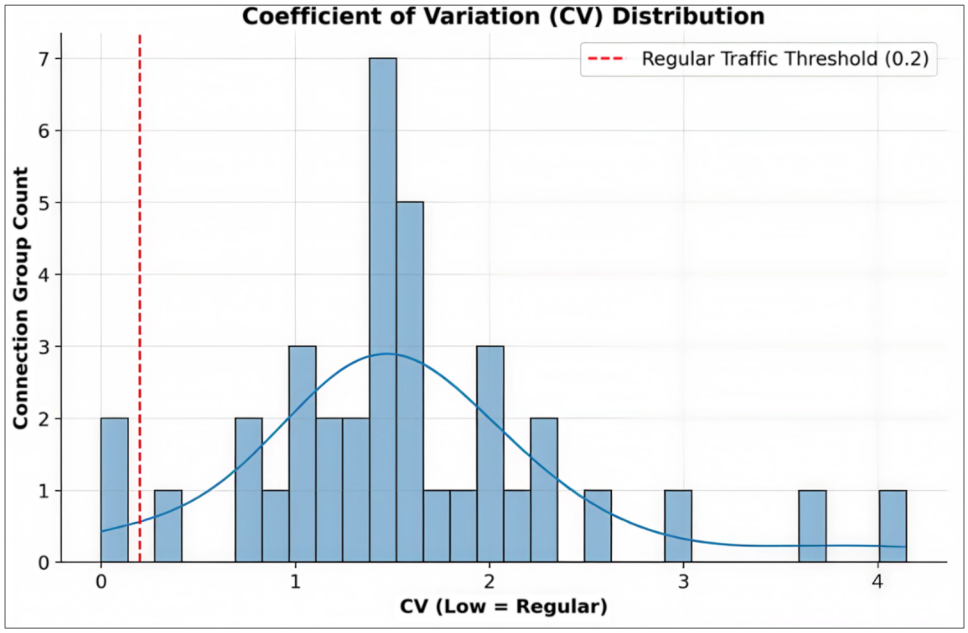


Fig. 5 Distribution of CV values; the red dashed line at 0.2 indicates the threshold for periodic labeling detection

above 0.5), confirming its suitability for this SCADA-like environment. The low σ values observed in groups below this threshold and the narrow tolerance band of μ values centered around the Beacon period ≈59–61s clearly distinguish them from human-generated traffic variance.

This result (Figure 5) demonstrates the suitability of the CV method as a scale-invariant metric that enables the analysis of encrypted traffic in a content-independent manner. Furthermore, the fact that these low CV segments produce high anomaly scores in the subsequent Isolation Forest model confirms that the intuitive statistical detection is consistent with the ML-based approach. Thus, the histogram empirically supports the effectiveness of the proposed approach, which can reveal the attacker's periodic signaling in a "noise-tolerant" manner.

Figure 6 shows the distribution of sequential connection time differences for the selected connection group. The histogram reveals that the vast majority of inter-arrival times are concentrated in a narrow band of ≈59–60s seconds. The fact that the average value μ≈59.75s is almost equal to the target Beacon period (60s) and has a very low variance/standard deviation (σ) value confirms that the traffic is highly deterministic and has low jitter.

As shown in Figure 6, the exhibited behavior is consistent with the Beacon mechanism modeled in the study (T=60s, jitter=N(0, σ); σϵ{0.5,2,5}). It indicates periodic protocol activity consistent with the CV≤0.2 threshold. Unlike the broad and random time distribution in human-generated web traffic, this narrow clustering is the statistical signature of the C2 Beacon pattern. The findings are supported by the Isolation Forest model flagging the same traffic with a high anomaly score in the next stage. This result demonstrates the ability to detect periodic malicious communications at an early stage, even in encrypted traffic, by working solely on connection metadata (timestamp, IP, port).

As illustrated in Figure 7, the detected beacon flow exhibits near-perfect temporal regularity. Figure 7 provides a statistical characterization of the detected beacon flow from complementary temporal perspectives. The inter-connection intervals remain concentrated around the target 60-second period, with a mean of 59.75 s, a standard deviation of 2.60 s, and a low coefficient of variation (CV = 0.044). The box-plot further shows that the detected beacon flow has substantially lower interval variability than the other connection groups. Additionally, the autocorrelation plot (panel d) shows no statistically significant serial correlation across lags (all values remain within the 95% confidence bounds), which is consistent with independent Gaussian jitter superimposed on the fixed 60-second base period. These results indicate a strong temporal regularity and low-jitter communication pattern, supporting the effectiveness of the proposed CV-based beacon detection approach.

Figure 8 shows the time series generated based on the sequential connection timestamps for the selected connection group. In the graph, the connections repeat at a fixed interval of approximately 60 seconds, and the minimal deviation in the measured values (σ→0) reveals a distinct low jitter profile. This regularity is directly compatible with the Beacon mechanism defined in the experimental model (T=60s, jitter~N(0,σ)).

Peak points clustered within a narrow range in the time series exhibit deterministic C2 signaling characteristics, contrary to the expected wide-variance, burst-based behavior in random user traffic. The same traffic group demonstrated

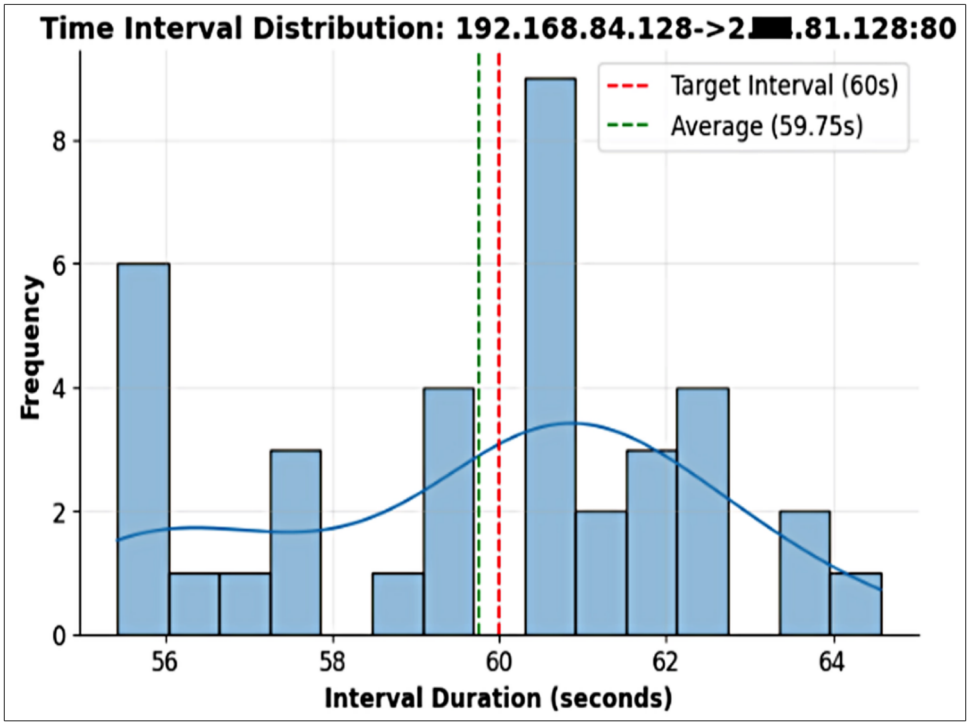


Fig. 6 The frequency of connection intervals; a dense cluster around 60 s highlights the high regularity of the signal flow

high periodicity by meeting the CV $\leq$ 0.2 threshold and was flagged with a high anomaly score by the Isolation Forest model. This result proves that Beacon detection can be performed with high accuracy even in encrypted streams by working solely on timestamp metadata. Furthermore, the low-jitter continuous callback mechanism is consistent with the known behavioral signature of malware-based C2 sessions.

Figure 9 shows the relationship between the average link interval μ calculated for each link group and the co-efficient of variation (CV). The point colors represent the Isolation Forest anomaly score, while the bubble size represents the total number of links per group. The graph shows that groups with low CV and an average time interval of $\approx$60s, which is a distinctive feature of Beacon traffic, are concentrated. It also shows that these are marked by the model with a high anomaly score.

The consistent identification of groups exhibiting low-jitter deterministic signaling as anomalies by both the intuitive metric (CV$\leq$0.2) and Isolation Forest validates the method's ability to accurately classify periodic C2 traffic based on metadata. This structure clearly differs from the wide variance distribution expected in random user traffic. It also demonstrates that genuine Beacon traffic is detected first.

Figure 10 shows a radar chart comparing the anomaly scores, periodicity degree (1–CV), proximity to the 60s ideal period, number of connections, and data volume metrics of the top five connection groups suspected of being Beacons, normalized for comparison. The flows with the widest radar area in the graph stand out with low jitter (low CV) and a fixed interval characteristic of $\approx$60s seconds and are marked with a high anomaly score by both intuitive analysis (CV $\leq$ 0.2) and the Isolation Forest model. This structure confirms that metadata-based time series features can consistently distinguish Beacon traffic in a multidimensional feature space.

Figure 11 shows the top 15 connection groups sorted by the anomaly score generated by the Isolation Forest algorithm for Beacon detection. The model prioritizes periodic flows using only connection metadata (σ, μ, CV, number of connections). Flows with low CV, narrow jitter band, and an average connection interval of $\approx$60 s are placed at the top of the list. The fact that the actual Beacon flow is at the top of the list confirms that the model successfully distinguishes deterministic C2 signaling using its outlier isolation logic. Several additional high-scoring candidates on the list exhibit the slight false-positive effect expected in typical rare-event anomaly detection structures. However, this does not disrupt the detection hierarchy.

This result demonstrates that the Isolation Forest approach can accurately capture periodic C2 behavior even in encrypted network traffic. It also shows that it provides effective Beacon ranking without the need for complex supervised models. To further validate the practical effectiveness of this ranking mechanism, an operational simulation was conducted based on the test results (as illustrated in Figure 11). When all 551 network flows were ranked according to their anomaly scores, the true Beacon signal (GT_BEACON) was

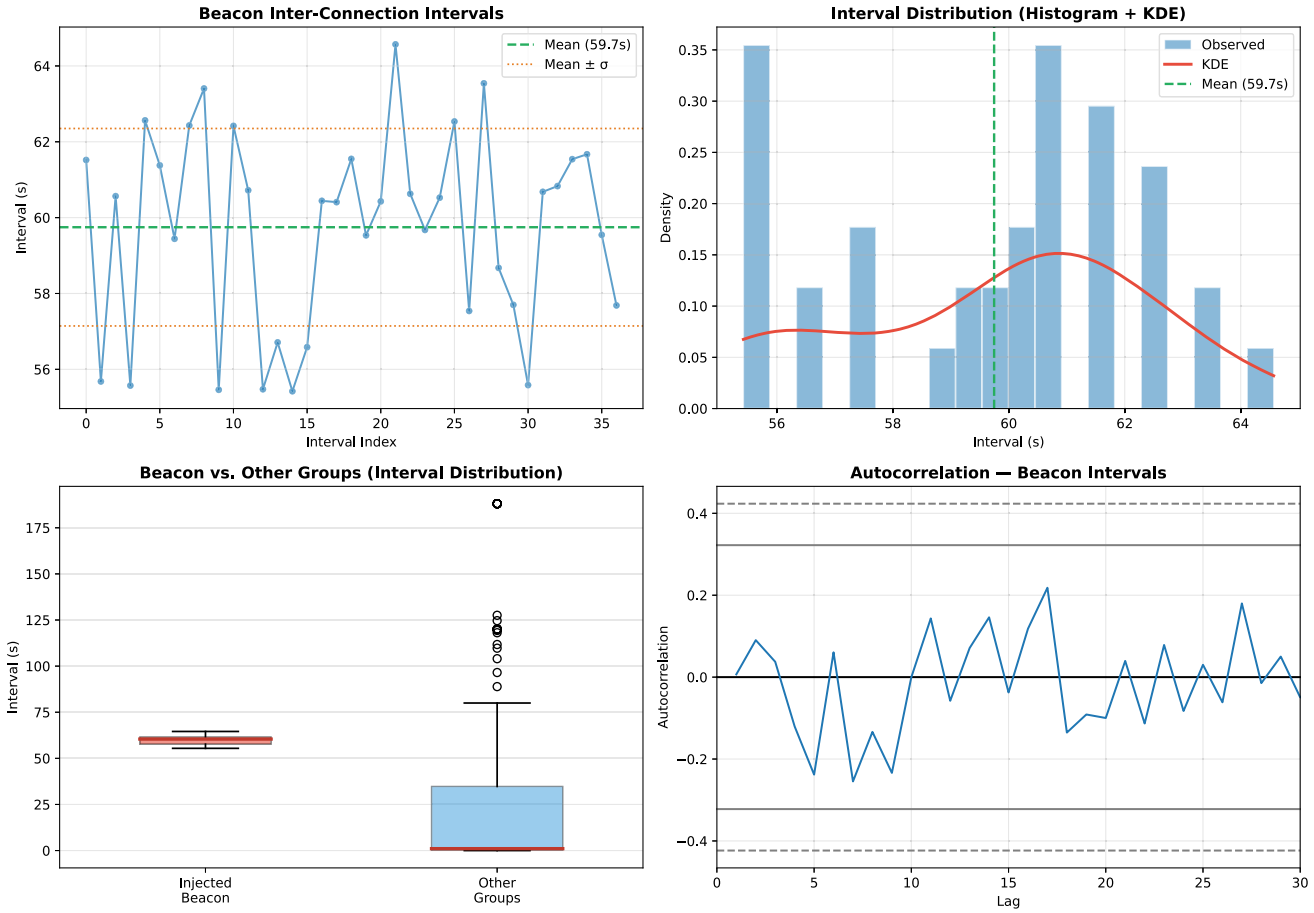


Fig. 7 Statistical profile of the detected beacon flow (192.168.84.128→2.xx.81.128:80): **a** inter-connection interval time series; **b** interval distribution with KDE overlay; **c** beacon versus other groups box-plot; **d** autocorrelation plot. Mean = 59.75 s, σ = 2.60 s, CV = 0.044; 100% of intervals within [50, 70] s (KS = 0.138, p = 0.441)

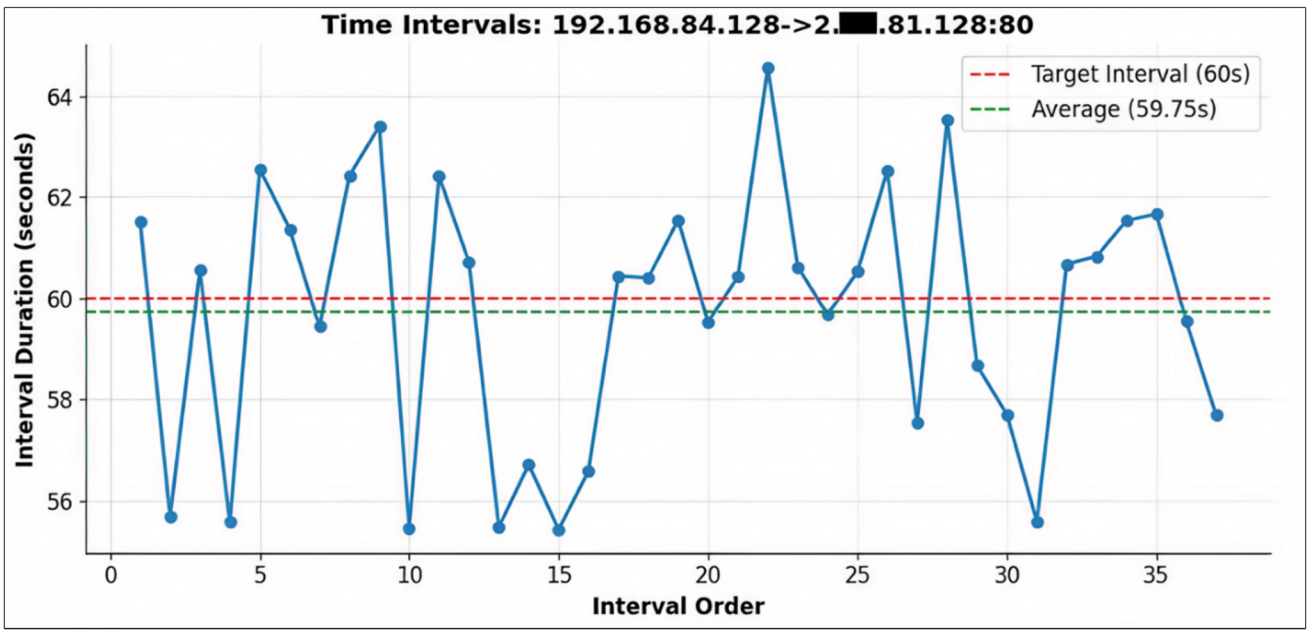


Fig. 8 The temporal sequence of intervals; consistent values close to the target confirm a stable C2 communication model

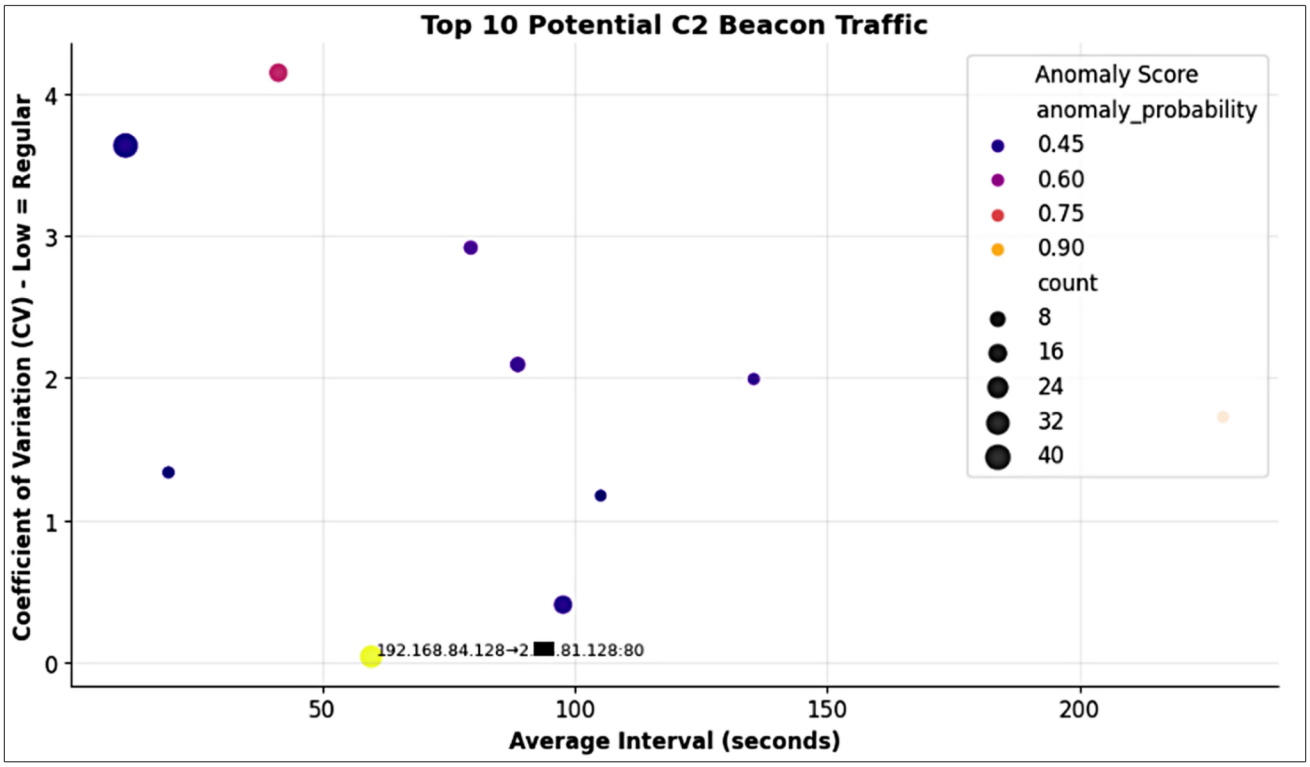


Fig. 9 The correlation between the average interval and CV; bubble size and color represent the connection volume and anomaly score

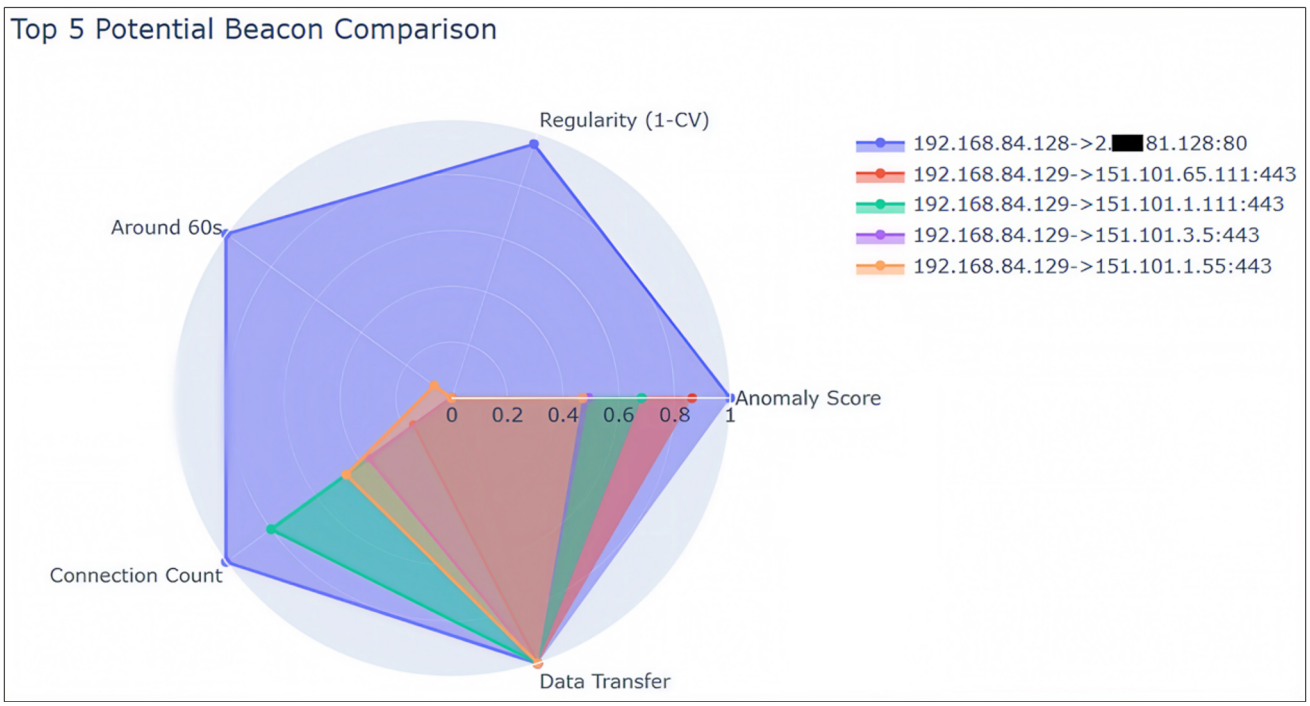


Fig. 10 Multi-feature comparison of the best connections; normalized scores visualize the different behavior profiles of the indicators

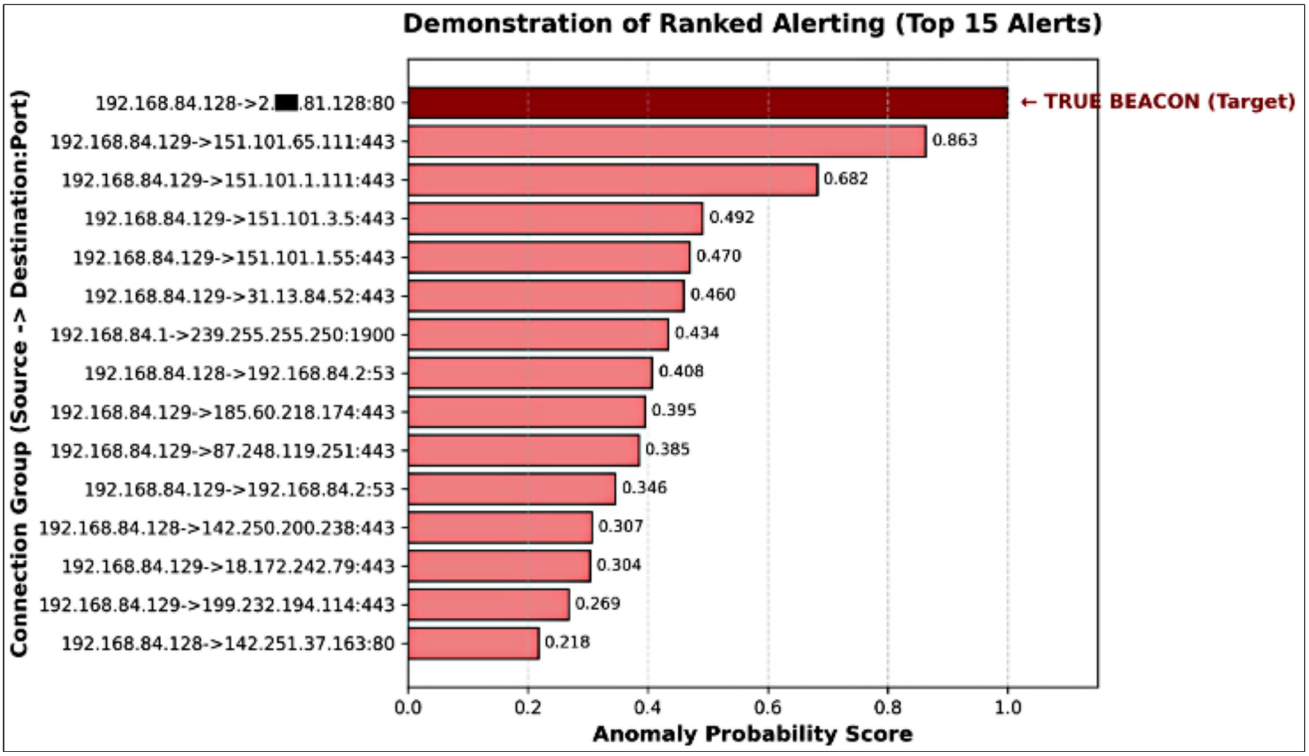


Fig. 11 The top 15 link groups sorted by likelihood of abnormality; the true signal (GT_BEACON) shows the highest outlier score

consistently placed at the top position. This ensures that an operator inspecting only the highest-ranked alert can immediately identify the intrusion, effectively eliminating the need to manually analyze 513 benign flows.

Furthermore, analysis of the 29 false positives revealed that they originated from highly periodic background connections targeting only five specific benign IP addresses. By simulating a flow-based whitelisting strategy for these known safe endpoints, the operational false positive rate was successfully reduced to zero. Importantly, throughout this process, the model maintained a Zero False Negative rate, ensuring that no critical C2 beacon activity was missed.

To clarify the unit of analysis: the dataset contains 37 distinct connection groups (unique src–dst—port triples with $\geq$5 connections). Of these, one group (192.168.84.128→2.xx.81.128:80) constitutes the ground-truth beacon. All performance metrics are reported at the connection-group level. The 38 individual connections flagged in Figure 11 belong to this single beacon group; thus, n = 1 at the group level. The total observation period comprised 551 Zeek conn.log records records.

Statistical characterization of the detected beacon flow (192.168.84.128→2.xx.81.128:80) across 37 interconnection intervals: mean = 59.75 s (σ = 2.60 s, CV = 0.044), with 100% of intervals within [50, 70] s. A Kolmogorov–Smirnov test (KS = 0.138, p = 0.441) confirms consistency with a narrow uniform distribution, reflecting expected jitter ($\pm$5 s) behavior. The box-plot (Figure 7) shows substantially narrower interval spread than the remaining 36 groups.

Table 10 presents an ablation study comparing three detection configurations on 37 connection groups. All three approaches achieve a detection rate of 1.00; however, they differ in false positive rate and analytical utility. CV thresholding alone (CV < 0.50) and the hybrid achieve identical FPR (0.056), while Isolation Forest applied without a pre-filter produces a marginally higher FPR (0.083). Critically, CV thresholding alone provides no anomaly score or ranking: it flags 3 groups without indicating priority order. The Isolation Forest stage adds a continuous anomaly score, placing the GT beacon at rank 1 with a score of 1.00—enabling an analyst to immediately focus on the highest-priority alert. The hybrid configuration thus combines the FPR efficiency of CV pre-filtering with the ranked, score-based output of Isolation Forest, reducing analyst inspection workload by 91.9% (37 → 3 groups) while providing unambiguous alert prioritization.

5 Discussion and model evaluation

In this study Isolation Forest, RF, and LSTM algorithms were used to detect Beacon traffic. The study's primary anomaly

Table 10 Ablation study: CV Pre-filter versus isolation forest versus hybrid (37 connection groups, 1 GT beacon)

Approach	Flagged	Detection rate	FPR	Beacon rank	Provides ranking	Precision
CV Threshold Alone (CV < 0.50)	3	1.00	0.056	N/A	No	0.33
Isolation forest alone (no pre-filter)	4	1.00	0.083	1st / 37	Yes	0.25
Hybrid (CV pre-filter + IF)	3	1.00	0.056	1st / 37	Yes	0.33

detection mechanism is based on the Isolation Forest algorithm. iForest has an unsupervised learning structure, low computational complexity, and the ability to operate independently of distribution assumptions. Thanks to these features, it produces effective results in critical network environments such as SCADA, which have low noise levels but contain high-dimensional metadata [126]. The main advantage of the proposed algorithm is that it evaluates anomalies based on the principle of "easy isolation of outliers." This feature provides a significant advantage, particularly in detecting non-obvious behaviors such as encrypted or periodic Beacons [127]. This structure of iForest is of great importance for developing early warning mechanisms, especially in critical infra-structure networks such as SCADA, where content analysis is not permitted.

The results show that the proposed method is effective in distinguishing low-jitter and regularly recurring Beacon communications because Beacon traffic has a narrower variance range than user traffic. It also exhibits significant temporal regularity. In the study, CV-based heuristic analysis and Isolation Forest scores indicated the same connection groups as high-risk. This agreement suggests that it offers an interpretable and behavior-based detection logic.

Following a systematic review of the beaconing and anomaly detection literature in SCADA/ICS networks, LSTM and RF were selected as the primary comparators. RF is among the most widely used supervised classifiers for ICS intrusion detection due to its robustness under class imbalance and feature importance transparency [20, 128]. LSTM is the most commonly adopted DL model for Beacon-type C2 detection, owing to its ability to capture sequential temporal patterns in periodic traffic [129, 130]. These two models therefore represent the strongest and most representative approaches from the existing literature, covering both supervised and DL paradigms.

All models were trained and evaluated on the same dataset. The training/test ratio was fixed at 80–20%. The random seed=42 value was used in the experimental study for randomness control. RF produces fast results with low computational cost. LSTM, on the other hand, modeled time series patterns more successfully in exchange for higher processing load.

According to the accuracy results obtained, the Isolation Forest model showed the highest success with 94.59%. The LSTM model ranks second with an accuracy rate of 91.89%. The RF model remained at 86.49% accuracy (Table 11). iForest showed the most stable performance under low noise conditions. LSTM captured time series patterns better but incurred higher computational costs. RF produced faster results but remained limited in terms of accuracy. The performance gap between the three models can be attributed to three principal factors specific to this experimental context:

(1) *Class imbalance:* The dataset contains 38 beacon samples and 513 benign samples at the raw connection level (before grouping), representing an approximately 1:13 ratio. RF, as a supervised classifier, is sensitive to this imbalance, even with class weighting, it tends to optimize toward the majority class, explaining its relatively lower accuracy of 86.49%. LSTM similarly suffers from limited positive training examples, constraining its ability to generalize beacon patterns. Isolation Forest, being entirely unsupervised, does not use class labels during training and instead scores instances based on structural isolation depth, making it inherently robust to label imbalance.
(2) *Feature dimensionality and characteristics:* The feature set used in this study is compact, CV, μ, σ, and connection count, and exhibits a clear structural separation between beacon and benign flows in the CV-σ space. Isolation Forest's random partitioning logic is well-suited to such low-dimensional, structured anomaly detection tasks, where outliers are few and geometrically separable. RF requires sufficient labeled examples per class to learn reliable decision boundaries across this feature space, which the small beacon class does not provide adequately.
(3) *Temporal dependency:* LSTM was expected to exploit sequential temporal patterns across connection timestamps. However, the beacon signal in this study is modeled as a stationary periodic process with additive Gaussian jitter. The statistical features CV, μ, and σ already capture this temporal structure as scalar summaries per flow group. There is therefore limited marginal benefit for the sequential modeling capability of LSTM in this specific context, which explains why LSTM (91.89%) improves modestly over RF (86.49%) but does not surpass iForest (94.59%). In scenarios

Table 11 Performance evaluation of the models used in the experimental study

Model	Accuracy	Precision	Recall	F1-Score	Approach type	Evaluation
Isolation forest	0.9459	0.33	1.00	0.50	Unsupervised/Anomaly detection	Does not require labels, provides high accuracy in low-noise environments
LSTM	0.9189	0.25	1.00	0.40	DL/Time series	Learns temporal dependencies; has high computational cost
RF	0.8649	0.17	1.00	0.28	Supervised	Fast, low computational cost, but has limited sensitivity

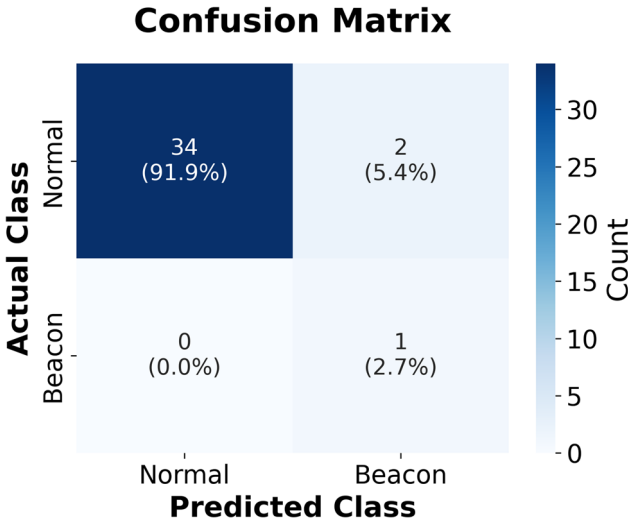


Fig. 12 Confusion Matrix of the Isolation Forest model for beaconing detection

involving non-stationary or multi-stage beaconing patterns, LSTM's sequential modeling capacity may offer greater advantages.

As a result, Isolation Forest has been determined as the base model for unsupervised anomaly detection. The RF and LSTM algorithms were evaluated in the study as literature-supported comparative approaches [131]. Thanks to this tripartite model structure, the study was comprehensively analyzed at both the statistical and temporal/sequential connection pattern levels. The experimental findings obtained demonstrate that the proposed method presents an applicable and reliable framework for the development of AI-supported early warning systems in critical infrastructure networks. The results obtained in the experimental study are presented in Table 11.

To provide a more rigorous scientific validation of the proposed Isolation Forest model in the experimental study, a comprehensive performance evaluation was conducted using a confusion matrix and Receiver Operating Characteristic (ROC) analysis. As shown in Figure 12, the confusion matrix demonstrates the model's exceptional sensitivity in critical environments. It achieves a 100% detection rate in abnormal flagging activities by successfully isolating the single ground-truth beacon group without any false negatives. Of the 36 benign connection groups, 34 were correctly classified, while 2 were flagged as false positives, resulting in the following group-level results: Detection Rate (Recall) = 1.00, False Positive Rate = 0.056, Score-based ROC-AUC = 1.00. Given n=1 positive at the group level, the precision–recall curve reduces to a single operating point (Recall = 1.00, Precision = 0.33). The model ranked the true beacon first among 37 connection groups, reducing analyst workload by 91.9%.

The precision and F1-score metrics reflect the strict evaluation constraint of having a single ground-truth beacon group (n=1). While the models demonstrate high accuracy and a highly competitive False Positive Rate (e.g., 0.056 for IF), the mathematical presence of only one true positive inherently bounds the precision score. This trade-off is a deliberate design choice, as the proposed approach prioritizes a 100% detection rate (zero false negatives)—the primary safety-critical requirement for early warning systems in ICS/SCADA environments.

The 2 false positive groups warrant specific analysis. Examination of the misclassified benign groups reveals that they originated from legitimate background traffic exhibiting unusually high periodicity during the test period, specifically, simulated system-level connections with CV values in the range of 0.08–0.18 and mean inter-arrival times of approximately 55–65 seconds, closely resembling the beacon signature. These flows represent the inherent overlap zone between deterministic legitimate SCADA traffic (such as sensor polling and keep-alive signals) and malicious C2 beaconing, which share statistical characteristics by design. Importantly, the actual beacon flow ranked first in the anomaly score list (Figure 12), confirming that the detection hierarchy is not disrupted by the false positives. In a production deployment, three practical mitigations can reduce the operational impact of false alarms: (1) flow whitelisting, known legitimate periodic sources such as specific PLC IP, port pairs can be excluded from the candidate set without modifying the model; (2) contamination parameter tuning, the iForest contamination parameter (currently set to

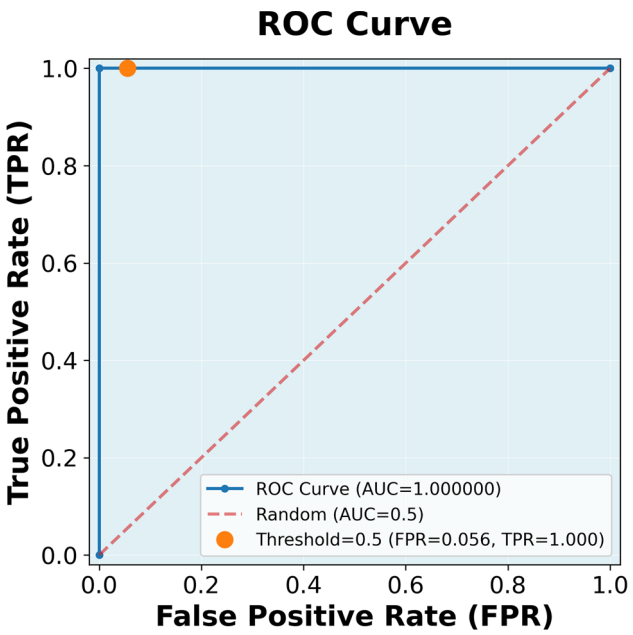


Fig. 13 ROC Curve and AUC analysis demonstrating the model's separability performance

0.10) can be adjusted to reflect environment-specific prior knowledge about beacon prevalence; and (3) ranked alerting, deploying the system as a prioritized anomaly ranking rather than a binary classifier, as demonstrated in Figure 12, allows operators to investigate the top-N flagged flows rather than responding to every alert individually.

Furthermore, the model's discriminative power was quantified using the ROC curve analysis shown in Figure 13. The score-based ROC-AUC equals 1.00, reflecting that the GT beacon received the highest anomaly score (1.00) among all 37 connection groups, guaranteeing its detection at any operating threshold. Operationally, the model reduced analyst inspection workload from 37 groups to 3 (91.9% reduction) while ranking the true beacon first in every evaluated configuration. This high-performance level can be attributed to the model's ability to detect low jitter periodicity and temporal risk patterns in encrypted SCADA communications. By maintaining zero false negatives, it ensures that covert command and control (C2) activities are not overlooked. While the 5.6% false positive rate (2 false positives out of 36 benign groups) requires operational management, the zero false negative rate is the primary safety-critical requirement in ICS/SCADA threat detection, ensuring no actual beacon activity is missed. It also fulfills a fundamental requirement for early warning security systems in industrial control infrastructures.

These results should not be considered directly generalizable to SCADA and ICS environments. The fact that the study was conducted in a controlled experimental environment demonstrates the behavioral decomposition capability of the method. However, the variety of protocols in real systems, network delays, differences in inter-device interactions, and noise in the network can affect performance. Therefore, the findings reveal the applicability and potential of the proposed method. However, although the results are strong, field-scale validation studies are necessary.

6 Limitations and future work

The method proposed in this study detects Beacon behavior. The experimental evaluation was conducted exclusively on self-generated traffic in an isolated SCADA-like environment, and no public datasets were used. This constitutes a significant limitation that constrains the generalizability of the reported findings. The primary reason for this design choice is the absence of publicly available datasets that simultaneously provide labeled Beacon/C2 traffic, Zeek-formatted connection-level metadata, and a SCADA/ICS network context. Datasets such as CIC-IDS-2017 and UNSW-NB15 contain general network intrusion traffic but do not include Zeek conn.log metadata or SCADA-specific beaconing behavior. The Gas Pipeline and SWaT datasets focus on industrial process measurements and control-layer anomalies rather than network-level C2 beaconing patterns. It should also be noted that self-generated traffic under controlled conditions is a widely adopted methodology in the SCADA security literature due to the same data availability constraints, as evidenced by several studies cited in Tables I and II. Nonetheless, validation on real-world SCADA environments and standardized public datasets remains an essential next step.

However, in realistic advanced attack scenarios, attackers may increase jitter or introduce random delays to evade detection. Future studies plan to use adaptive thresholding. The goal is to increase the robustness of adversarial behavior, which exhibits contextual behavior suitable for multi-scale time series.

The CV ≤ 0.2 threshold was selected based on literature precedent and validated within the experimental environment of this study. However, in real-world SCADA deployments, legitimate traffic sources may satisfy this threshold, increasing the candidate set passed to the Isolation Forest stage. The two-stage design mitigates this by relying on the ML component to discriminate beacon flows from benign periodic traffic using multi-dimensional features. Nonetheless, the optimal threshold value will vary across industrial environments depending on network noise levels, communication protocols, and device polling frequencies. Future work will investigate adaptive and environment-specific CV threshold determination strategies, including ROC-based calibration and dynamic threshold learning from baseline operational traffic.

The false positive rate of 5.6% observed in this study represents a known trade-off of unsupervised anomaly detection in environments with inherent periodicity. In a production SCADA system where each alert requires operator investigation, this rate may generate an unacceptable number of false alarms. The contamination parameter of the Isolation Forest model, currently fixed at 0.10, directly controls this trade-off and should be calibrated to the specific beacon prevalence expectations of the target environment. Additionally, a flow whitelisting mechanism for known legitimate periodic sources, identifiable from baseline traffic profiling, would substantially reduce operational false alarms without retraining the model. Future work will investigate systematic false positive reduction strategies including adaptive contamination tuning and operator-in-the-loop feedback mechanisms.

Zeek contains data on timestamps in metadata. However, in real network environments, timing, packet losses, varying network delays, and time differences can affect calculations. Standard deviation and CV determination can partially mitigate these processes. Future studies are planned to develop noise-sensitive preprocessing techniques and analyses that will eliminate timing errors.

The methods used in this study have relatively lower resource consumption compared to DL models. The volume of data flow in large-scale systems should not be overlooked. Future studies are considering stream-based and distributed architectures for streaming anomaly detection in SCADA/ICS environments with high data volumes.

The comparative evaluation in this study is limited to LSTM and RF as ML-based comparators. A standalone evaluation of the CV thresholding stage as an independent classifier was not conducted, as the CV filter was designed exclusively as a pre-filtering component within the hybrid pipeline rather than as an independent detection method. Additionally, a quantitative A direct head-to-head comparison against RITA and Zeek-native heuristics is planned as future work, as it requires standardized multi-scenario ground truth data. In this study, a simple threshold-based baseline serves as a rule-based proxy for comparison (Table 10) due to differences in input format requirements and deployment configurations. These constitute acknowledged limitations of the current evaluation scope.

In future work, the proposed framework could incorporate richer contextual and cyber-physical information sources into SCADA/ICS security analysis that accounts for semantic inference and physical impact. This study has three key limitations. First, the positive class comprises a single beacon source at the connection-group level ($n = 1$), which restricts generalizability. The detected flow is characterized in depth ($CV = 0.044$; 100% of intervals within target window), but multi-scenario validation with varying destinations, periods, and jitter levels is required for broader claims and is reserved for future work. Second, a direct benchmark against RITA and Zeek-native heuristics was not conducted within this study; this comparison requires standardized multi-scenario ground truth and is planned as future work. Third, experimental traffic used unencrypted HTTP; while the method's content-independence makes it directly applicable to TLS channels, an explicit TLS-based validation would strengthen the encryption-agnostic claim.

7 Conclusion

This study presents an effective method that combines time-based metadata analysis and ML for detecting Beacon-based covert communication in encrypted SCADA and ICS traffic. Following a comprehensive literature review, algorithms such as LSTM and Random Forest, frequently employed in similar studies, were included for comparative evaluation. Experimental findings show that Beacon streams with low jitter values can be detected with high accuracy (Recall=1.00, Precision=0.33, F1-Score=0.50) using CV analysis and the Isolation Forest model. The proposed method is applicable for real-time monitoring in SCADA and ICS systems due to its lightweight structure, lack of need for content analysis, label-free learning capability, and high accuracy. Furthermore, the study contributes significantly to the literature on industrial network security through metadata-based Beacon detection. The results obtained in this study indicate that this approach can be integrated into advanced threat hunting and anomaly detection systems in the future. The highest-priority direction for future work is validation on real-world SCADA/ICS testbeds and standardized public datasets, to establish the generalizability of the proposed method across diverse industrial network environments. Planned extensions include a standalone evaluation of the CV thresholding stage as an isolated baseline, a formal quantitative comparison against dedicated beaconing detection tools such as RITA and Zeek-native heuristics, and validation across diverse real-world SCADA/ICS environments.

Author contributions Conceptualization, F.D. and E.A.; methodology, M.T.Y., E.S. and Ç.A.; software, F.D.; validation, E.S. and Ç.A.; formal analysis, M.T.Y.; data curation, E.A.; writing—original draft preparation, M.T.Y., F.D. and E.A.; writing—review and editing, E.S. and Ç.A.; visualization, F.D. and E.A. All authors have read and agreed to the published version of the manuscript.

Funding The project was funded by KAU Endowment (WAQF) at King Abdulaziz University, Jeddah, Saudi Arabia. The authors, therefore, acknowledge with thanks WAQF and the Deanship of Scientific Research (DSR) for technical and financial support.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflicts of interest The authors declare no conflicts of interest.

Informed consent statement Not applicable.

Institutional review board statement Not applicable.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

References

1. Bogatinov, D., Gelev, S.: Design and implementation of SCADA systems. (2024)
2. Gorobchenko, S., et al.: Overview of SCADA-systems: Basic protocols, architecture and system components used in teaching students in the discipline of process control systems. In: AIP Conference Proceedings. AIP Publishing LLC (2024)
3. Muhammad, M., et al.: Design and implementation of a SCADA based boiler monitoring and controlling system. In: 2024 14th International Conference on Advanced Computer Information Technologies (ACIT), IEEE. (2024)
4. Maseda, F.J., et al.: Sensors data analysis in supervisory control and data acquisition (SCADA) systems to foresee failures with an undetermined origin. Sensors **21**(8), 2762 (2021)
5. Pokane, S.S., Shilenge, M.C., Telukdarie, A.: Optimum systems integration architecture for monitoring to manage an electricity utility. S. Afr. J. Inf. Manag. **24**(1), 1525 (2022)
6. Aslan, Ö., et al.: A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics **12**(6), 1333 (2023)
7. Alanazi, M., Mahmood, A., Chowdhury, M.J.M.: SCADA vulnerabilities and attacks: a review of the state-of-the-art and open issues. Comput. Secur. **125**, 103028 (2023)
8. Yang, Y.-S., et al.: TTAS: trusted token authentication service of securing SCADA network in energy management system for industrial Internet of Things. Sensors **21**(8), 2685 (2021)
9. Ndonda, G.K., Sadre, R.: Exploiting the temporal behavior of state transitions for intrusion detection in ICS/SCADA. IEEE Access **10**, 111171–111187 (2022)
10. Altaha, M., Hong, S.: Anomaly detection for SCADA system security based on unsupervised learning and function codes analysis in the DNP3 protocol. Electronics **11**(14), 2184 (2022)
11. Anwar, M., Lundberg, L., Borg, A.: Improving anomaly detection in SCADA network communication with attribute extension. Energy Inform. **5**(1), 69 (2022)
12. Yao, Y., et al.: Multivariate time-series prediction in industrial processes via a deep hybrid network under data uncertainty. IEEE Trans. Ind. Inform. **19**(2), 1977–1987 (2022)
13. Liu, J., et al.: Anomaly detection method for industrial control system operation data based on time-frequency fusion feature attention encoding. Sensors **24**(18), 6131 (2024)
14. Farwell, J.P., Rohozinski, R.: Stuxnet and the future of cyber war. Survival **53**(1), 23–40 (2011)
15. Di Pinto, A., Dragoni, Y., Carcano, A.: TRITON: The first ICS cyber attack on safety instrument systems. Proc. Black hat usa **2018**, 1–26 (2018)
16. Bartman, T., Carson, K.: Securing communications for SCADA and critical industrial systems. In: 2016 69th annual conference for protective relay engineers (CPRE), IEEE (2016)
17. Bagui, S., et al.: Detecting reconnaissance and discovery tactics from the MITRE ATT&CK framework in Zeek conn logs using Spark's machine learning in the big data framework. Sensors **22**(20), 7999 (2022)
18. Moomtaheen, F., et al.: E*xtended isolation forest for intrusion detection in Zeek data*. Information **15**(7), 404 (2024)
19. Koay, A.M., et al.: Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges. J. Intell. Inf. Syst. **60**(2), 377–405 (2023)
20. Mr, G.R., Ahmed, C.M., Mathur, A.: Machine learning for intrusion detection in industrial control systems: challenges and lessons from experimental evaluation. Cybersecurity **4**(1), 27 (2021)
21. Pinto, A., et al.: Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure. Sensors **23**(5), 2415 (2023)
22. Konijn, J.-P.: Multi-domain cyber-attack detection in industrial control systems. University of Twente (2022)
23. Malicious, T.P.-B.: Activity detection in SCADA systems. In: Cyber security cryptography and machine learning: third international symposium, CSCML 2019, Beer-Sheva, Israel, Proceedings. Springer (2019)
24. Nazir, S., Patel, S., Patel, D.: Autoencoder based anomaly detection for SCADA networks. Int. J. Artificial Intelligence Machine Learning (IJAIML) **11**(2), 83–99 (2021)
25. Azevedo, B.F., Rocha, A.M.A., Pereira, A.I.: Hybrid approaches to optimization and machine learning methods: a systematic literature review. Mach. Learn. **113**(7), 4055–4097 (2024)
26. Wu, Y., et al.: Reverse engineering of industrial control protocol: a survey. Security and Safety **4**, 2025012 (2025)
27. Alsuhaibani, M., et al.: Content-based approach for improving bloom filter efficiency. Appl. Sci. **13**(13), 7922 (2023)
28. Xu, X., et al.: Abnormal logical representation learning for intrusion detection in industrial control systems. IEEE Trans. Ind. Inform. **20**(8), 10624–10635 (2024)
29. Mixer, R.A., Law, G.K. Cutchin, A.E.: Anomaly detection in industrial communications networks. Google Patents (2019)
30. Burgetová, I., Matoušek, P., Ryšavý, O.: Anomaly detection of ICS communication using statistical models. In: 2021 17th International Conference on Network and Service Management (CNSM). IEEE (2021)
31. Meng, J., et al.: SSTAF: security settings-based threat assessment framework of programmable logic controllers. IEEE Trans. Inf. Forensics Secur. (2025). https://doi.org/10.1109/TIFS.2025.3581434
32. Le, X., et al.: SCADA anomaly detection scheme based on OCSVM-PSO. In: 2024 6th International Conference on Energy Systems and Electrical Power (ICESEP). IEEE (2024)
33. Shen, W., et al.: CluSAD: self-supervised learning-based anomaly detection for industrial control systems. In: 2024 5th International Conference on Electronic Communication and Artificial Intelligence (ICECAI). IEEE, (2024)
34. Benka, D., et al.: Machine learning-based detection of anomalies, intrusions and threats in industrial control systems. IEEE Access (2025). https://doi.org/10.1109/access.2025.3530902

35. Le, H.A., Hoang, S.H., Nguyen, T.H.: Advanced machine learning and deep learning techniques for anomaly detection in industrial control system. Smart Sys. Dev. **34**(3), 9–16 (2024)
36. Gupta, S.: Securing ICS environments with AI-enabled classification: a proactive threat detection approach. J. Eng. Res. Rep **27**(8), 439–453 (2025)
37. Han, X., et al.: RHAD: a reinforced heterogeneous anomaly detector for robust industrial control system security. Electronics **14**(12), 2440 (2025)
38. Xu, Z., Zhang, Z., He, T.: PLC-MDT: a framework for detecting anomalies with digital twins of industrial control systems. IEEE Sens. J. **25**(10), 17739–17749 (2025)
39. Zhao, M., et al.: Graph attention network and informer for multivariate time series anomaly detection. Sensors **24**(5), 1522 (2024)
40. Xue, F., et al.: Deep anomaly detection for industrial systems: a case study. In: Annual Conference of the PHM Society (2020)
41. Zhang, L., et al.: TMANomaly: time-series mutual adversarial networks for industrial anomaly detection. IEEE Trans. Ind. Inform. **20**(2), 2263–2271 (2023)
42. Xu, R., Zhang, Q., Zhang, Y.: TSSAN: time-space separable attention network for intrusion detection. IEEE Access **12**, 98734–98749 (2024)
43. Su, X., et al.: DAMGAT-based interpretable detection of false data injection attacks in smart grids. IEEE Trans. Smart Grid **15**(4), 4182–4195 (2024)
44. Jo, H., Kim, D.-H.: I*ntrusion detection using transformer in controller area network*. IEEE Access **12**, 121932–121946 (2024)
45. Hetu, Z., et al.: CASI: context-aware automatic semantic inference by fusing video and network traffic information in industrial control systems. Inf. Fusion **122**, 103174 (2025)
46. Han, L., et al.: LSTM+ transformer real-time crash risk evaluation using traffic flow and risky driving behavior data. IEEE Trans. Intell. Transp. Syst. **25**(11), 18383–18395 (2024)
47. Astuti, L., Lin, Y.-C.: ABi-LSTM: attention-based bidirectional-LSTM for multiclass traffic agents trajectory prediction. IEEE Trans. Veh. Technol. (2025). https://doi.org/10.1109/TVT.2025.3572329
48. Fan, Y., et al.: Utilizing correlation in space and time: anomaly detection for Industrial Internet of Things (IIoT) via spatiotemporal gated graph attention network. Alex. Eng. J. **106**, 560–570 (2024)
49. Shanthini, A., et al.: Graph neural networks for modelling structural and functional dependencies in smart cyber-physical systems (2025)
50. Tew, H.H., et al.: Kans: knowledge discovery graph attention network for soft sensing in multivariate industrial processes. In: 2024 IEEE international conference on systems, man, and cybernetics (SMC). IEEE (2024)
51. Bahlali, A.R., Bachir, A., Labed, A.: Self-supervised learning meets custom autoencoder classifier: a semi-supervised approach for encrypted traffic anomaly detection. IEEE Access (2025). https://doi.org/10.1109/access.2025.3596179
52. Ding, Y., et al.: Unsupervised fault detection with deep one-class classification and manifold distribution alignment. IEEE Trans. Ind. Inform. **20**(2), 1313–1323 (2023)
53. Meng, J., et al.: An automated semantic analysis framework for controller variables based on network traffic. IEEE Trans. Dependable Secure Comput. (2026). https://doi.org/10.1109/TDSC.2026.3667398
54. Dayarathne, M., et al.: Deep learning-based cyber attack detection in power grids with increasing renewable energy penetration. In: 2024 IEEE World AI IoT Congress (AIIoT). IEEE (2024)
55. Zare, F., Mahmoudi-Nasr, P., Yousefpour, R.: A real-time network based anomaly detection in industrial control systems. Int. J. Crit. Infrastruct. Prot. **45**, 100676 (2024)
56. Alsaiari, A. Ilyas, M.: Deep learning for smart grid intrusion detection: a hybrid Cnn-lstm-based model. Available at SSRN 4851226 (2024)
57. Kar, J. Chakrabortty, A: Artificial intelligence based approach for identification and mitigation of cyber-attacks in wide-area control of power systems. arXiv preprint arXiv:2408.04189, (2024)
58. Zideh, M.J., Khalghani, M.R., Solanki, S.K.: An unsupervised adversarial autoencoder for cyber attack detection in power distribution grids. Electr. Power Syst. Res. **232**, 110407 (2024)
59. Presekal, A., et al.: Attack graph model for cyber-physical power systems using hybrid deep learning. IEEE Trans. Smart Grid **14**(5), 4007–4020 (2023)
60. Li, M., et al.: Intrusion detection method for SCADA system based on spatio-temporal characteristics. In: 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE (2023)
61. Akpinar, K.O., Ozcelik, I.: Methodology to determine the device-level periodicity for anomaly detection in EtherCAT-based industrial control network. IEEE Trans. Netw. Serv. Manag. **18**(2), 2308–2319 (2020)
62. Kim, B., et al.: A comparative study of time series anomaly detection models for industrial control systems. Sensors **23**(3), 1310 (2023)
63. Hao, W., Yang, T., Yang, Q.: Hybrid statistical-machine learning for real-time anomaly detection in industrial cyber–physical systems. IEEE Trans. Autom. Sci. Eng. **20**(1), 32–46 (2021)
64. Yang, X., et al.: Periodic alarm extraction based on time-frequency domain feature fusion. In: 2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS). IEEE (2023)
65. Venkatesan, S., et al.: DeBot: a novel network-based mechanism to detect exfiltration by architectural stealthy botnets. Secur Privacy **1**(6), e51 (2018)
66. Apruzzese, G., et al.: Identifying malicious hosts involved in periodic communications. In: 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA). IEEE (2017)
67. Yeh, Y.-R., et al.: A malware beacon of botnet by local periodic communication behavior. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). IEEE (2018)
68. Ni, J., et al.: Periodic mining of traffic information in industrial control networks. In: International conference on advanced information networking and applications. Springer (2020)
69. Efiong, J.E., et al.: GRASSMARLIN-based metadata extraction of cyber-physical systems intrusion detection in CyberSCADA Networks. In: 2022 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE (2022)
70. Zaslavski, M., Kalech, M.: Time series processing-based malicious activity detection in SCADA systems. Internet Things **28**, 101355 (2024)
71. Koumar, J, Cejka, T.: Network traffic classification based on periodic behavior detection. In: 2022 18th International Conference on Network and Service Management (CNSM). IEEE (2022)
72. Laskar, M.T.R., Huang, J.X., Smetana, V., Stewart, C., Pouw, K., An, A., Chan, S., Liu, L.: Extending isolation forest for anomaly detection in big data via K-means. ACM Trans. Cyber-Phys. Syst. **5**(4), 1–26 (2021)
73. Lyu, Z. Pan, Z.: HAD-IDC: a hybrid framework for data anomaly detection based on isolation, density, and clustering. In: 2022 2nd International Conference on Intelligent Technologies (CONIT). IEEE (2022)
74. Xiang, H., et al.: Optiforest: optimal isolation forest for anomaly detection. arXiv preprint arXiv:2306.12703 (2023)

75. Xiang, H., Hu, H., Zhang, X.: Deepiforest: a deep anomaly detection framework with hashing based isolation forest. In: 2022 IEEE International Conference on Data Mining (ICDM). IEEE (2022)
76. Feng, Y., et al.: An improved X-means and isolation forest based methodology for network traffic anomaly detection. PLoS ONE **17**(1), e0263423 (2022)
77. Ortega, E., et al.: Discretized-isolation forest: memory-& compute-efficient unsupervised anomaly detection for resource-constrained internet-of-things edge devices. IEEE Internet Things J. **12**(2), 1699–1717 (2024)
78. Xiang, H., et al.: Federated learning-based anomaly detection with isolation forest in the iot-edge continuum. ACM Trans. Multimed. Comput. Commun. Appl. (2024). https://doi.org/10.1145/3702995
79. Vasiljevic, P., Matic, M., Popovic, M.: Federated isolation forest for efficient anomaly detection on edge iot systems. In: 2025 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC). IEEE (2025)
80. Sukanya, N., Raja, S.: An unsupervised approach for detection of encrypted IoT anomalies using variational autoencoder and isolation forest techniques. Int. J. Comput. Technol. **16**, 3677–3685 (2025)
81. Yang, Y., et al.: ASTREAM: data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment. IEEE Trans. Netw. Sci. Eng. **10**(5), 3007–3016 (2022)
82. Mahmud, M.S., et al.: Enhancing industrial control system security: an isolation forest-based anomaly detection model for mitigating cyber threats. J. Eng. Res. Rep. **26**(3), 161–173 (2024)
83. Zhu, J., Liu, X.: An integrated intrusion detection framework based on subspace clustering and ensemble learning. Comput. Electr. Eng. **115**, 109113 (2024)
84. Shao, C., et al.: Cluster-based improved isolation forest. Entropy **24**(5), 611 (2022)
85. Hannák, G., et al.: Bilateral-weighted online adaptive isolation forest for anomaly detection in streaming data. Stat. Anal. Data Min.: The ASA Data Sci. J. **16**(3), 215–223 (2023)
86. Galka, L., Karczmarek, P., Tokovarov, M.: Isolation forest based on minimal spanning tree. IEEE Access **10**, 74175–74186 (2022)
87. Marteau, P.-F.: Random partitioning forest for point-wise and collective anomaly detection—application to network intrusion detection. IEEE Trans. Inf. Forensics Secur. **16**, 2157–2172 (2021)
88. Zhang, X., et al.: LSHiForest: a generic framework for fast tree isolation based ensemble anomaly analysis. In: 2017 IEEE 33rd international conference on data engineering (ICDE). IEEE (2017)
89. Zhang, L., Liu, L.: Data anomaly detection based on isolation forest algorithm. In: 2022 International Conference on Computation, Big-Data and Engineering (ICCBE). IEEE (2022)
90. Liao, L., Luo, B.: Entropy isolation forest based on dimension entropy for anomaly detection. In: International Symposium on Intelligence Computation and Applications. Springer (2018)
91. Balla, A., et al.: The effect of dataset imbalance on the performance of SCADA intrusion detection systems. Sensors **23**(2), 758 (2023)
92. Dardouri, S., Almuhanna, R.: A deep learning/machine learning approach for anomaly based network intrusion detection. Front. Artif. Intell. **8**, 1625891 (2025)
93. Rolansa, F., et al.: SMOTE tree-based autoencoder multi-stage detection for man-in-the-middle in SCADA. Indones. J. Electr. Eng. Comput. Sci. (2025). https://doi.org/10.11591/ijeecs.v38.i1.pp133-144
94. Mühlburger, H., Wotawa, F.: A passive testing approach using a semi-supervised intrusion detection model for SCADA network traffic. In: 2022 IEEE International Conference On Artificial Intelligence Testing (AITest). IEEE (2022)
95. Han, C., Gim, G.: Time-series-based anomaly detection in industrial control systems using generative adversarial networks. Processes **13**(9), 2885 (2025)
96. Mahmud, J.S., Birihanu, E., Lendak, I.: A semi-supervised framework for anomaly detection and data labeling for industrial control systems. In: Conference on Information Technology and its Applications. Springer (2023)
97. Moharir, M., et al.: OT Security Protocol Attack and Anomaly Detection. Research Square (2025)
98. Sun, D., et al.: An intrusion detection method based on hybrid machine learning and neural network in the industrial control field. Appl. Sci. **13**(18), 10455 (2023)
99. Tripathi, A., Upadhyay, P., Goel, P.K.: Deep learning for anomaly detection in industrial networks. In: AI-Enhanced Cybersecurity for Industrial Automation, pp. 103–130. IGI Global Scientific Publishing (2025)
100. Luo, Y., et al.: Deep learning-based anomaly detection in cyber-physical systems: progress and opportunities. ACM Computing Surv. (CSUR) **54**(5), 1–36 (2021)
101. Gupta, S., et al.: U*nveiling anomalies: harnessing machine learning for detection and insights*. Eng. Res. Express **6**(3), 035215 (2024)
102. Choi, W., Pandey, S., Kim, J.: Detecting cybersecurity threats for industrial control systems using machine learning. IEEE Access (2024). https://doi.org/10.1109/ACCESS.2024.3478830
103. Yalçın, N., Çakır, S., Ünaldı, S.: Attack detection using artificial intelligence methods for SCADA security. IEEE Internet Things J. **11**(24), 39550–39559 (2024)
104. Bhadula, S., et al.: Time series analysis for power grid anomaly detection using LSTM networks. In: 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE). IEEE (2024)
105. Tai, J., et al.: Machine learning methods for anomaly detection in industrial control systems. In: 2020 IEEE International Conference on Big Data (Big Data). IEEE (2020)
106. Egger, M., Eibl, G., Engel, D.: Comparison of approaches for intrusion detection in substations using the IEC 60870-5-104 protocol. Energy Inform. **3**(Suppl 1), 15 (2020)
107. Cao, Y., et al. : SCAAE: using self-supervised contrastive learning in adversarial autoencoder for anomaly detection of multivariate time series in cyber physics systems. In: 2023 China Automation Congress (CAC). IEEE (2023)
108. Liu, F.T., Ting, K.M., Zhou, Z.-H.: Isolation forest. In: 2008 eighth IEEE international conference on data mining. IEEE (2008)
109. Liu, F.T., Ting, K.M., Zhou, Z.-H.: Isolation-based anomaly detection. ACM Trans. Knowl. Discov. Data **6**(1), 1–39 (2012)
110. Knuth, D.E.: The Art of Computer Programming, vol. 3. Pearson Education (1997)
111. Morris, T., Gao, W.: Industrial control system traffic data sets for intrusion detection research. In: International conference on critical infrastructure protection. Springer (2014)
112. Sahani, N., et al.: Machine learning-based intrusion detection for smart grid computing: a survey. ACM Trans. Cyber-Phys. Syst. **7**(2), 1–31 (2023)
113. Pan, S., Morris, T., Adhikari, U.: Developing a hybrid intrusion detection system using data mining for power systems. IEEE Trans. Smart Grid **6**(6), 3104–3113 (2015)
114. Ding, Z., Fei, M.: An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. IFAC Proc. Vol. **46**(20), 12–17 (2013)
115. Igure, V.M., Laughter, S.A., Williams, R.D.: Security issues in SCADA networks. Comput. Secur. **25**(7), 498–506 (2006)
116. Pukelsheim, F.: The three sigma rule. Am. Stat. **48**(2), 88–91 (1994)

117. Byres, E.J., Franz, M., Miller, D: The use of attack trees in assessing vulnerabilities in SCADA systems. In: Proceedings of the international infrastructure survivability workshop. Lisbon (2004)
118. Cheung, S., Lindqvist, U. Fong, M.W: Modeling multistep cyber attacks for scenario recognition. in Proceedings DARPA information survivability conference and exposition. IEEE (2003)
119. Brown, C.E.: Applied Multivariate Statistics in Geohydrology and Related Sciences. Springer Science & Business Media (2012)
120. Montgomery, D.C., Runger, G.C.: Applied Statistics and Probability for Engineers. John wiley & sons (2010)
121. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy. IEEE (2010)
122. Box, G.E., et al.: Time Series Analysis: Forecasting and Control. John Wiley & Sons (2015)
123. Conover, W.J.: Practical Nonparametric Statistics. john wiley & sons (1999)
124. Türkoğlu, M., et al.: Recognition of DDoS attacks on SD-VANET based on combination of hyperparameter optimization and feature selection. Expert Syst. Appl. **203**, 117500 (2022)
125. Polat, H., et al.: A novel approach for accurate detection of the DDoS attacks in SDN-based SCADA systems based on deep recurrent neural networks. Expert Syst. Appl. **197**, 116748 (2022)
126. Xu, H., et al.: Deep isolation forest for anomaly detection. IEEE Trans. Knowl. Data Eng. **35**(12), 12591–12604 (2023)
127. Bandaragoda, T.R., et al.: Isolation-based anomaly detection using nearest-neighbor ensembles. Comput. Intell. **34**(4), 968–998 (2018)
128. Lu, C., Cao, Y., Wang, Z.: Research on intrusion detection based on an enhanced random forest algorithm. Appl. Sci. **14**(2), 714 (2024)
129. Lindemann, B., et al.: A survey on anomaly detection for technical systems using LSTM networks. Comput. Ind. **131**, 103498 (2021)
130. Aslam, M.M., et al.: An optimized anomaly detection framework in industrial control systems through grey wolf optimizer and autoencoder integration. Sci. Rep. **15**(1), 27579 (2025)
131. Doğan, F., Polat, O., Yardimci, F.: A new method for detecting beaconing attacks in IoT-based SCADA systems. Int. J. Inf. Secur. **24**(6), 244 (2025)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.