IoT-GRAF: IoT Graph Learning-based Anomaly and Intrusion Detection through Multi-modal Data Fusion

Rozhin Yasaei[1], Yasamin Moghaddas[2], Mohammad Abdullah Al Faruque[2]
University of Arizona[1], University of California Irvine[2]
yasaei@arizona.edu, ymoghadd@uci.edu, alfaruqu@uci.edu

***Abstract*—In the current technological landscape, Internet of Things (IoT) systems are deeply embedded in numerous facets of daily life, from domestic settings to critical infrastructure, which underscores the importance of these systems security and integrity. The constrained nature of IoT devices, in terms of computational capacity, economic limitations, or time-to-market, makes them vulnerable to security breaches and system failures. Additionally, the hybrid essence of IoT- combining the physical domain via sensor interfaces and the cyber domain through communication networks and cloud connectivity- further complicates mitigating these threats. While numerous techniques for either network intrusion detection or sensor anomaly detection exist, an integrated approach that synergistically combines information from both domains is absent. This paper proposes a multi-modal data fusion technique, which, for the first time, melds sensor and communication data. This approach underscores the interdependencies between the components, provides contextual embeddings for data from each element, and integrates the system's physical and cyber features into a graph-based representation. Harnessing the power of Graph Neural Networks (GNNs), we capture the normal state and context of the system, facilitating the detection of anomalies and intrusions. Additionally, our model discerns between network and sensor-based attacks, pinpointing the anomaly's origin, thereby expediting post-incident recovery. Optimized for fog-computing environments, our solution ensures real-time oversight. Rigorous testing on greenhouse IoT systems indicates the efficacy of our model, with a commendable 22% improvement in F1-score over singular modal techniques.**

***Index Terms*—Internet of Things, sensor fusion, anomaly detection, graph neural network, security, multi-modal, heterogeneous**

I. Introduction

The Internet of Things (IoT) comprises a vast interconnected network of sensors and devices used across various sectors, with the potential for a global economic impact of up to $11.1 trillion by 2025. Despite their growth and importance, the low cost and limited computational capability of many IoT devices make them vulnerable to attacks. Ensuring IoT security is paramount, given its extensive use in critical applications. While many strategies exist to detect anomalies in time-series sensor data and network intrusion, they often overlook the interconnected context of IoT systems. Our study introduces a holistic, real-time approach for multi-modal data fusion that combines sensor and communication data analysis to detect both anomalies and attacks, filling the existing gap in the literature.

A. Motivational Example

Wireless device users in outdoor settings can experience varied connectivity depending on the day and environmental conditions. Research has explored the influence of factors like temperature and humidity on wireless communication. Specifically, [1] found that temperature adversely affects the received signal strength indicator (RSSI), with humidity being particularly influential at temperatures below $0°C$. The study also highlighted notable RSSI variations across individual channels and links, possibly due to multipass propagation. Although deriving an exact model linking environmental factors and RSSI might be challenging, statistical models can depict a correlation, especially in low-power wireless networks. This example highlights the necessity to incorporate sensor data as well as network information to determine an attack/anomaly and its cause. Relying only on network information, the RSSI variation in the example could be interpreted as network intrusion.

B. Research Challenges and Opportunities

In IoT anomaly detection, researchers grapple with challenges stemming from the vast volumes of time-series data sensors generate. Key issues include dimensionality reduction, ensuring minimal loss of crucial feature information, and managing complications like a dimensional explosion and concept drift [2]. Over time, accumulating noisy data can distort anomaly predictions while evolving standards of what is normal further complicate detection accuracy. This complexity amplifies with multivariate time-series data. Furthermore, the unique interactions in IoT as a cyber-physical system foster mutual knowledge among system nodes. Deriving insights from this context can be invaluable for understanding system behavior and pinpointing abnormalities. However, embedding this topology and context into a model remains intricate.

C. Our Contribution

In summary, our contributions are summarized as follows:

- We propose a novel holistic approach to the security and integrity of IoT systems. It performs selective sensor and network data fusion to detect anomalies and attacks on the system's communication network and physical layout.
- We extract the shared context among multi-modal components of the system through time-series data analysis, and we selectively fuse only related data in the data fusion and machine learning processes.
- We propose a heterogeneous graph representation for IoT systems that embodies sensing devices and communication network parameters as nodes and correlation between component pairs as a connection.

979-8-3503-4859-0/DATE24/© 2024 EDAA

- We leverage GNN to automatically extract key features of the system from its graph representation and detect anomalous activities.

II. Related Works and Background

This paper closely correlates with three well-established research areas discussed in this section.

A. Network Intrusion Detection

Network Intrusion Detection (NID) employs abnormality detection techniques to identify deviations in network traffic patterns. With the rise of deep learning, numerous models have been introduced to enhance NID. D-PACK [3] combines CNN and AE for traffic categorization while facing memory challenges. Kim et al. [4] designed a CNN model tailored for Denial-of-Service (DoS) detection, surpassing earlier RNN designs, while another of their works [5] utilizes local outlier factor and AE for malware detection. E-GraphSage [6] integrates a GNN model, emphasizing network topology, but has space and time complexity issues. In terms of physical cyberattacks, Qiu et al. [7] uses an adaptive neural network examining channel characteristics, and Liao et al. [8] offers a hybrid method for physical layer authentication in wireless sensors, emphasizing efficiency in communication resources.

B. Data Fusion and Anomaly Detection

In recent decades, machine learning has become a prominent tool for data fusion and time-series analysis. CNNs and RNNs are often employed for anomaly detection due to their ability to handle both spatial and temporal information. Yin et al. [9] developed a model integrating CNN and recurrent autoencoder for univariate time-series anomaly detection, emphasizing temporal relevance through a two-stage sliding window. Meanwhile, Zhang et al. [10] introduced a Dual-Window RNN-CNN for periodic multivariate time-series data, leveraging both the GRU to learn temporal features and a CNN-based Autoencoder to determine feature dependencies. While effective, their datasets had limited anomalies. [11] addressed multivariate time-series anomalies using a deep convolutional clustering-based model, enhancing detection performance through K-means clustering loss in the AE latent space.

C. GNN for anomaly detection

Machine learning has emerged as the preferred approach for data fusion and time-series analysis in recent years. CNNs and RNNs play pivotal roles due to their ability to handle spatial and temporal information for anomaly detection. Yin et al. [9] combined CNN and recurrent autoencoders, focusing solely on univariate time-series data, while Zhang et al. [10] introduced a Dual-Window RNN-CNN to handle the complexities of periodic multivariate data, capturing temporal and spatial dependencies with a CNN-based Autoencoder and multi-head GRU, albeit with datasets having few anomalies. Meanwhile, Chadha et al. [11] targeted multivariate time-series anomaly detection using a deep one-dimensional CNN autoencoder, enhancing performance by partitioning the AE latent space and employing a K-means clustering loss.

III. Methodology: Multi-modal Data Fusion

We aim to integrate physical and network layers of IoT systems and detect security threats and malfunctions. Our methodology is explained in this section according to the pipeline depicted in Figure 1.

A. IoT Network Security Analysis

This section delves into prevalent attacks against the Long Range Wide Area Network (LoRaWAN), used in IoT systems like greenhouse monitoring [12]. We examine various attack impacts and their distinctive signatures on communication metadata such as RSS and SNR. Given IoT system's low power and cost constraints, these networks are susceptible to several cyberattacks, including spoofing, jamming, replay, and wormhole attacks [13]. **Spoofing:** Here, attackers pose as genuine nodes to corrupt or alter data. Specifically, LoRaWAN is sensitive to acknowledgment spoofing [14]. Significant RSS fluctuations might indicate an attack, as devices typically maintain consistent transmission power. **Jamming attacks:** These attacks, a subset of DoS attacks, involve intentional signal interference [15]. While generic jamming impacts all devices on a given frequency, selective jamming targets individual devices, making its detection challenging. [16] introduced a selective jamming mechanism effective against LoRaWAN packets, achieving a 98% success rate. Jamming's primary aim is to elevate the SNR, where a decreased SNR indicates potential jamming, leading to frequent packet losses [17]. **Replay attack:** Attackers intercept and resend messages, allowing data and metadata manipulation. Sung et al. [18] devised a method to shield LoRaWAN devices from these attacks, utilizing RSSI and handshaking. A constantly shifting RSSI might hint at an ongoing replay attack. **Wormhole attacks:** Here, an attacker's device seizes a data packet and transfers it to another malicious device for multiple replays. This type of attack allows metadata tampering, like RSSI, SNR, and packet travel time. A study by [19] revealed that these attacks can be executed using wormholes, increasing RSSI and SNR.

B. Anomaly Implementation

In our work, we simulate various security incidents (detailed in Section III-A) in IoT systems incorporating synthetic anomalies in 10% of the data. To understand the signatures of these threats and gauge their effects on sensor and network data.

Network Anomalies: For each node, anomalies were inserted at random data points (timestamps). The type of anomaly was also randomly selected. To simulate spoofing, the RSSI was augmented with an added data anomaly. RSSI values ranged between -40dBm and -30dBm. Packet dropping was simulated for replay attacks, and time lags representative of packet drop intervals were added. Wormhole attacks involved modifications to RSSI, SNR, and humidity and temperature readings. RSSI values were selected from high (-40dBm to -30dBm) or low (-100 dBm to -90 dBm) ranges. SNR was reduced to represent increased noise, with values ranging between -30dB and -20dB. Lastly, jamming was simulated by decreasing SNR and introducing packet drops.

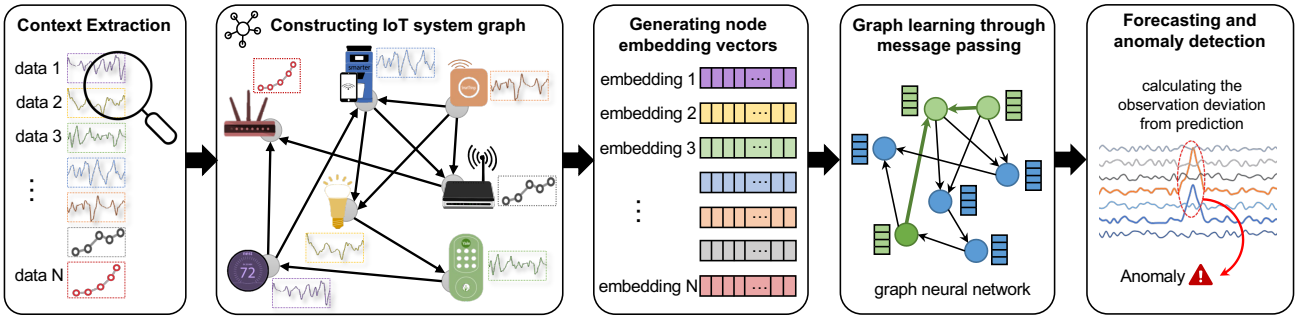


Fig. 1. Multi-modal data fusion pipeline for anomaly detection in IoT systems.

Sensor Anomalies: Sensor measurement anomalies are either global or contextualized outliers [20]. Global outliers deviate significantly from other values but still lie within the sensor's operational range. Upper and lower bounds for these outliers, *upper_outlier* and *lower_outlier*, are computed from the interquartile range. Anomalies are randomly selected from the ranges (*smallest_global_value*, *lower_outlier*) and (*upper_outlier*, *largest_global_value*). Contextualized outliers deviate from the average of a specific range of values. For a time instance X_t, the context is given by the values in the range (X_{t-l}, X_{t+l}), where l represents context length. The anomalous value, A, is derived as $A = \overline{X}_{t-l,t+l} + (\lambda * \sigma)$ with λ denoting the contextual threshold and σ representing the standard deviation of the sub-sequence values.

C. Data Preprocessing

In the preprocessing phase of our multivariate time-series dataset, we focused on ensuring data periodicity, continuity, and synchronization. Upon analyzing the data, each sensor shared the same periodicity, albeit with some missing data gaps. We establish a threshold for allowable time differences, targeting values greater than the sensor period but within the threshold. Achieving synchronization requires us to make the data continuous to ensure uniform timestamp counts across all sensors. This involved timestamp normalization to one-minute intervals, handling outliers, and aligning start and end time.

D. Context Extraction and Graph Generation

GNN is designed to process data structured as a graph. This accommodates non-Euclidean data within deep learning frameworks. The basic graph structure is represented as $G = (V, E)$, where V denotes nodes (in this context, sensors) and E the edges (relationships between sensors). We visualize our IoT system, especially for a greenhouse, as a graph, with nodes symbolizing system components and edges defining the correlation between these components. Contextual relationships between the components, vital for modeling the IoT system, are encoded as edges, guiding the graph learning procedure by merging related data. The relationships between nodes are stored in an adjacency matrix, A_{ij}. These relationships might be provided manually based on system understanding or extracted using data analysis. For automatic extraction, we experiment with cosine similarity and a correlation matrix. By default, sensors are interrelated, making A_{ij} a matrix of 1's. If specific sensors aren't interrelated, the corresponding matrix entry is 0. For component data similarity, we use cosine similarity for node embeddings, as shown:

$$e_{ji} = \frac{x_i^T x_j}{|x_i| \cdot |x_j|} \tag{1}$$

$$A_{ji} = \begin{cases} 1, & \text{if} j \in Top - K(\{e_{ki} : k \in S_i\}) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Here, S_i is the set of embeddings excluding x_i. The adjacency matrix, A_{ji}, is constructed from cosine similarities. The model uses the Top-K indices during its forward propagation. Alternatively, the similarity between sensor embeddings can be determined using the correlation coefficient matrix, $R_{ji} = \frac{S_{ji}}{\sqrt{S_{jj} S_{ii}}}$ where S represents the covariance matrix. The adjacency matrix is assembled similarly to the cosine similarity method.

E. Forecasting Graph Neural Network

Before we carry out anomaly detection, we need a baseline to compare data and find anomalies. We train our GNN model using a forecast-based approach to predict the values at a future time based on a window of values from the past. The input to the model, *x(t)*, *N*X*w*, where *N* is the number of nodes in the dataset, and *w* is the size of the sliding window:

$$x^t := s^{(t-W)}, s^{(t-W+1)}, \cdots, s^{(t-1)} \tag{3}$$

The output predicts the node values at time $s(t)$. The aforementioned learned graph is then used to perform graph attention-based feature extraction. It aggregates a node's features with the information of its neighboring node through a process called message passing. We aggregate sensor *i*'s input feature vector, $x_{i^{(t)}}$ with all of the features of its neighboring nodes to produce z_i:

$$\mathbf{z}_i^{(t)} = ReLU\left(\alpha_{i,i}\mathbf{W}\mathbf{x}_i^{(t)} + \sum_{j \in N_{(i)}} \alpha_{i,j}\mathbf{W}\mathbf{x}_j^{(t)}\right), \tag{4}$$

Where $N_{(i)} = \{j \ |\mathrm{A}_{ji} > 0\}$ is the neighborhood of sensor i. $W \in \mathrm{R}_{d \times w}$. α_{ij} are attention coefficients and can be found with the following calculations:

$$\mathbf{g}_i^{(t)} = \mathbf{v}_i \oplus \mathbf{W}\mathbf{x}_i^{(t)} \tag{5}$$

TABLE I
STATISTICS OF THE GREENHOUSE AND SWAT DATASET

Dataset	#devices/ features	Train	Test	Anomalies
Greenhouse	110	10128	2533	10%
SWaT	51	47515	44986	12%

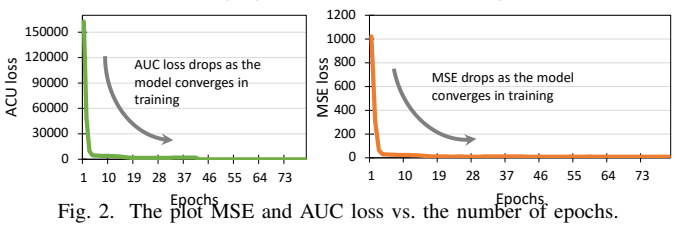


Fig. 2. The plot MSE and AUC loss vs. the number of epochs.

$$\pi\left(i,j\right)=LeakyReLU\left(\mathbf{a}^{T}\left(\mathbf{g}_{i}^{(i)}\oplus\mathbf{g}_{i}^{(j)}\right)\right) \tag{6}$$

$$\alpha_{i,i}=\frac{exp\left(\pi\left(i,j\right)\right)}{\sum_{k\in N_{(i)}\cup\{i\}}exp\left(\pi\left(i,k\right)\right)} \tag{7}$$

W is a trainable weight matrix that performs a linear transformation on a node's input feature vector, $\mathbf{x}_i^{(t)}$. $\mathbf{g}_i^{(t)}$ concatenates the sensor embedding and the transformed result. LeakyReLU calculates the attention coefficient, after which it is normalized with a softmax function. Each aggregated sensor, z_i, is then multiplied element-wise with its embedding, v_i. The final results for each sensor are concatenated and represent the vector of predicted values, $\hat{s}^{(t)}$ for each sensor at time t:

$$\hat{s}^{(t)}=f_\theta\left(\left[\boldsymbol{v_i}\circ\boldsymbol{z_i^{(t)}},\ldots,\boldsymbol{v_N}\circ\boldsymbol{z_N^{(t)}}\right]\right) \tag{8}$$

F. Anomaly Detection

Our model is trained on an anomaly-free training dataset to establish a baseline for normal behavior and tested against a testing dataset. Anomaly detection is performed in real-time on a validation dataset, differentiating our model from traditional deep learning models that test in batches. We calculate the deviation from the normal state, considering the trade-off between precise anomaly detection and minimizing false positives. Several strategies are employed to determine these deviations:

Gaussian Estimator Method: Error scores between actual and predicted values are computed. Time-series data's seasonality is accounted for by assigning different thresholds for different times of day. Gaussian distributions, based on test dataset losses for each time of day, are then used to calculate data point probabilities using the probability density function. Thresholds are determined based on potential candidates ranging from p_{min} to p_{max}, optimizing for the highest F1-score.

Standard Deviation Method: This method employs mean μ and standard deviation σ of the test results to define anomalies. If any data value at time t deviates from the normal range, it's labeled as an anomaly.

$$state(v_t)=\begin{cases}1, & \text{if } v_t<\mu-2.5*\sigma\\ 1, & \text{else if } v_t>\mu+2.5*\sigma\\ 0, & \text{otherwise}\end{cases} \tag{9}$$

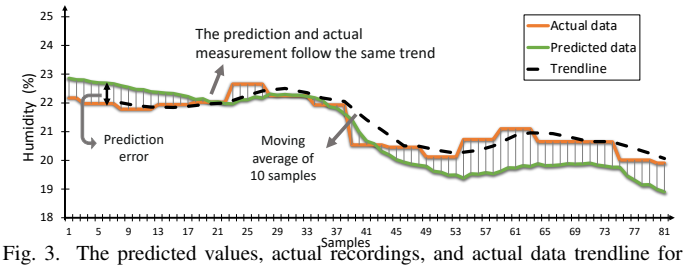


Fig. 3. The predicted values, actual recordings, and actual data trendline for a humidity sensor follow a similar pattern.

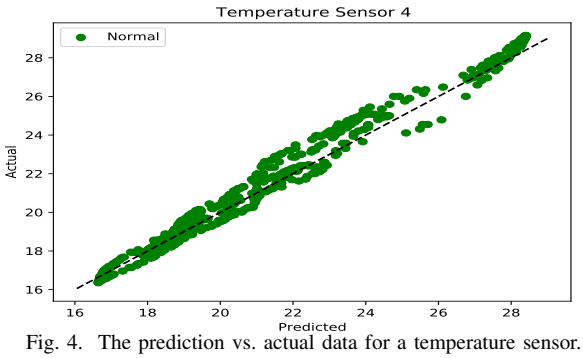


Fig. 4. The prediction vs. actual data for a temperature sensor.

Error Score Method: Error scores at time t for each test data are calculated as $Err_i(t)=|s(t)_i-\hat{s}(t)_i|$. These scores are normalized:

$$a_i(t)=\frac{Err_i(t-\widetilde{\mu}_i)}{\widetilde{\sigma}_i} \tag{10}$$

Where $\widetilde{\mu}_i$ and $\widetilde{\sigma}_i$ represent the median and inter-quartile range of each sensor's error array across time t. Anomalies are detected by comparing each sensor's error score to the threshold.

Max Node Prediction Score Method: Like the previous method, but uses the maximum of all node prediction scores at t to determine anomalies. These techniques allow the model to dynamically adapt to different conditions and effectively identify real-time anomalies in the data.

IV. EVALUATION

To assess our multi-modal data fusion approach, we studied data from a LoRaWAN-based greenhouse IoT system monitoring tomato crops in Belgium. This dataset, containing communication and sensor data, includes 22 devices logging temperature, humidity, RSSI, and SNR over five months. The extracted communication delay led to a total of 110 components, represented as a system graph with 110 nodes, each with 12661 data instances. The dataset was divided into 80% for training (normal data) and 20% for testing and validation, with the latter having 10% anomalies. Our main approach, the multi-modal model, utilizes the entire greenhouse dataset with 110 nodes. To examine the benefits of fusing varied data types and integrating communication with sensor data, we also created single-modal models. This single-modal technique resulted in 5 distinct models, each focusing on a specific sensor type with 22 nodes. Additionally, we employed the public Secure Water Treatment (SWaT) dataset, which contains normal operation data and attack scenarios, to compare the performance of our GNN anomaly detection model against other existing methods.

TABLE II
DATA FORECASTING ERROR AND GRAPH CHARACTERISTICS FOR DIFFERENT MODELS

Methodology	Total loss (MSE)	#nodes	#edges
Multi-modal	0.29	110	5500
Only Temperature	0.16	22	132
Only Humidity	12.12	22	132
Only SNR	0.25	22	220
Only RSS	0.78	22	51
Only Delay	0.84	22	51

All methods and variants were implemented using PyTorch version 1.5.1, CUDA 10.2, and the PyTorch Geometric library.

A. Forecasting Model Performance

Our pipeline begins with data preprocessing, context extraction, and graph generation to transform the data into a graph representation. This is followed by our GNN model, trained on an anomaly-free dataset to establish a normal behavior baseline. The Mean Squared Error (MSE) loss function given by:

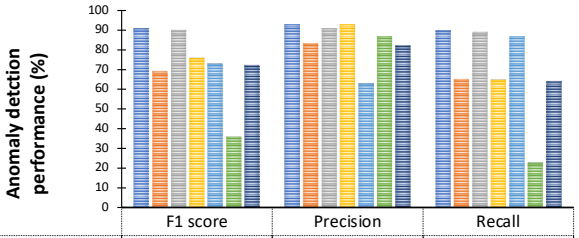
$$L_{MSE} = \frac{1}{T_{train} - w} \sum_{t=w+1}^{T_{train}} \left(\hat{s}^{(t)} - s^{(t)} \right)^2 \qquad (11)$$

Measures prediction errors. As training progresses, the loss reduces until changes become negligible, signifying model convergence. This loss reflects the model's proficiency in learning time-series patterns and predicting node values. Visualizations, like Figures 3 and 4, demonstrate the model's prediction accuracy. Table II reveals that multi-modal data fusion outperforms single-modality methods, underscoring the efficacy of our GNN-based approach in merging data from varied system layers for improved learning and model precision.

B. Anomaly Detection Performance

The forecasting model predicts the upcoming data instance based on historical data, which is the expected normal value. The deviation from expectation is analyzed upon observation of actual data to determine whether an anomaly has occurred and label the data instance. We use F1-score, precision, and recall as evaluation metrics to assess anomaly detection performance.

Single-modal vs. Multi-modal Anomaly Detection: Testing the single-modal and multi-modal models on anomaly-infested testing dataset resulted in the performance reported in Figure 5. The results indicate that the multi-modal approach outperforms the single-modal approach on average by 22%. Although for some data types, such as temperature, the single-modal model has comparable results to multi-modal, standalone models fail to detect various anomalies because different attacks have varied impacts and signatures, as discussed in Section III-A. All the parameters are required to capture diverse types of anomalies, and on average, anomaly detection on a single data modality performs inferior. The underlying reason the multi-modal dataset performs much better than the single-modal datasets is that its graph is more comprehensive. Hence, it provides more context during attention-based forecasting. The graph density in Table II is the total number of edges for each dataset, which ends up being the total number of nodes used for training multiplied by the top k values of the normalized dot



	F1 score	Precision	Recall
Multi-modal	91	93	90
Ave. single-modal	69	83	65
Temperature	90	91	89
Humidity	76	93	65
SNR	73	63	87
RSS	36	87	23
Delay	72	82	64

Fig. 5. The performance of multi-modal Vs. single-modal models.

TABLE III
COMPARING CONTEXT EXTRACTION METHODS: COSINE SIMILARITY VS. CORRELATION COEFFICIENT MATRIX

Algorithm	F1-score	Precision	Recall
Cosine Similarity	91%	93%	90%
Correlation Coefficient Matrix	83%	87%	79%

products computed during graph generation. The multi-modal method graph has about five times as many edges as the single-modal datasets, which could challenge the training process but embeds more detailed information.

Context Extraction Evaluation: We investigate the impact of the initial similarity extraction algorithm on the model performance and compare the performance of the multi-modal method using the cosine similarity versus the correlation coefficient matrix to generate the adjacency matrix. In Table III, the results show that cosine similarity supersedes the correlation coefficient matrix with higher F1-score, precision, and recall.

Comparing Anomaly Detection Methods in the Literature: We compare the GNN-based approach for anomaly detection in time-series data with other popular techniques in the literature. We base the comparison on the SWaT dataset and gather the results of different methods. Table IV compares models performance as well as the characteristics of each model. The comparison reveals that GNN outperforms others.

C. Timing Analysis

Effective IoT system supervision mandates swift, real-time anomaly detection, even with the constrained computational resources characteristic of IoT. Relying on cloud servers for model execution results in significant data transmission overhead as raw measurements travel from edge devices through the fog layer to the cloud. Hence, our model is optimized for real-time anomaly detection on a fog platform. A key advantage of our multi-modal fusion technique is being more compact than the aggregate of single-modal models, reducing both computational overhead and memory usage. Table V showcases the training, testing, and validation times for the data, all executed on an X86-64 CPU. Training and testing times are one-time operations for setting up the system. The subsequent validation time represents the real-time detection latency. The real-time validation is conducted per data instance, ensuring timely monitoring. Ideally, the processing time for each data point should stay below the system's data collection frequency. Our

TABLE IV
COMPARING ANOMALY DETECTION PERFORMANCE OF STATE-OF-THE-ART METHODS FOR SWAT DATASET.

Method	F1-score	Precision	Recall
GNN	81%	99%	68%
MAD-GAN [21]	77%	99%	63%
LSTM-VAE [22]	74%	96%	60%
AE [23]	61%	52%	72%
DAGMM [24]	39%	27%	70%
PCA [25]	23%	25%	22%
FB [26]	10%	10%	10%
KNN [27]	8%	8%	8%

TABLE V
TIMING OF TRAIN, TEST, AND VALIDATION FOR EACH DATA POINT.

Methodology	Training time	Testing time	Validation time
Multi-modal	150ms	5.9ms	3.7ms
Only Temperature	67ms	3.8ms	2.4ms
Only Humidity	66ms	6.0ms	2.7ms
Only SNR	67ms	4.0ms	2.5ms
Only RSS	65ms	3.7ms	2.7ms
Only Delay	67ms	3.7ms	2.6ms

approach achieves an impressive 3.7ms anomaly detection time, meeting real-time requirements for our greenhouse system.

D. Attack Analysis

Our method deeply integrates system communication and sensing nodes through multi-modal fusion, allowing for detailed analysis of anomalies and tracing them to their source. As detailed in Section III-A, we explore the consequences of various cyber and physical attacks. Physical attacks target sensors with faulty data, leading to abnormal measurements, while attacks like denial of service may result from system failures or physical breaches, causing data irregularities.

Conversely, communication channel attacks affect both cyber readings, such as SNR or RSSI, and the physical layer. For instance, Figure 6 showcases a discrepancy between actual and predicted readings from temperature, humidity sensors, and their communication channel SNR. Notable data changes, while other system components remain normal, hint at potential abnormal activities. After identifying an attack signature, we can ascertain the nature of the threat, such as a spoofing attack, and adapt our response accordingly. Similar methods apply to other attacks like jamming, replay, and wormhole.

V. CONCLUSION

In this study, we introduce a method for the selective fusion of sensor and communication data to detect anomalies in IoT systems in real-time. Utilizing a graph representation with GNN, we learn system behaviors and trace anomalies back to their origins and types of attack. This not only aids in system recovery but also guides appropriate security measures. To our understanding, this is the pioneering work in using GNN for fusing sensor and communication data, paving the way for more integrated and accurate digital modeling of real-world systems.

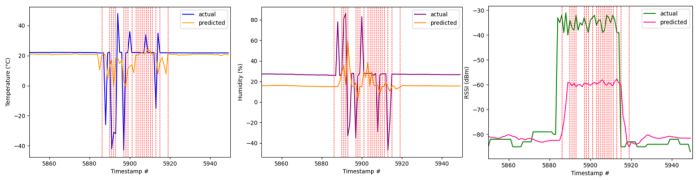


Fig. 6. The impact of a spoofing attack on temperature, humidity, and RSSI readings (left to right).

REFERENCES

[1] J. Luomala and I. Hakala, "Effects of temperature and humidity on radio signal strength in outdoor wireless sensor networks," in *Federated Conference on Computer Science and Information Systems*, 2015.

[2] Z. e. a. Chen, "Deep learning based anomaly detection for muti-dimensional time series: A survey," in *China Cyber Security Annual Conference*, 2021.

[3] R.-H. e. a. Hwang, "An unsupervised deep learning model for early network traffic anomaly detection," *IEEE Access*, 2020.

[4] J. e. a. Kim, "Cnn-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, no. 6, p. 916, 2020.

[5] S. e. a. Kim, "Anomaly based unknown intrusion detection in endpoint environments," *Electronics*, vol. 9, no. 6, p. 1022, 2020.

[6] W. W. e. a. Lo, "E-graphsage: A graph neural network based intrusion detection system," *arXiv preprint arXiv:2103.16329*, 2021.

[7] X. e. a. Qiu, "A learning approach for physical layer authentication using adaptive neural network," *IEEE Access*, 2020.

[8] R.-F. e. a. Liao, "Deep-learning-based physical layer authentication for industrial wireless sensor networks," *sensors*, 2019.

[9] C. e. a. Yin, "Anomaly detection based on convolutional recurrent autoencoder for iot time series," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[10] S. e. a. Zhang, "Anomaly detection of periodic multivariate time series under high acquisition frequency scene in iot," in *2020 International Conference on Data Mining Workshops (ICDMW)*, 2020.

[11] G. S. e. a. Chadha, "Deep convolutional clustering-based time series anomaly detection," *Sensors*, 2021.

[12] R. K. e. a. Singh, "Joint communication and sensing: A proof of concept and datasets for greenhouse monitoring using lorawan," *Sensors*, 2022.

[13] R. e. a. Gurunath, "An overview: security issue in iot network," in *2018 2nd International Conference on I-SMAC*, 2018.

[14] X. Yang, "Lorawan: Vulnerability analysis and practical exploitation," *Delft University of Technology. Master of Science*, 2017.

[15] H. Pirayesh and H. Zeng, "Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2022.

[16] E. e. a. Aras, "Selective jamming of lorawan using commodity hardware," in *International Conference on Mobile and Ubiquitous Systems*, 2017.

[17] M. M. R. e. a. Monjur, "An attack analysis framework for lorawan applied advanced manufacturing," in *2021 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2021.

[18] W.-J. e. a. Sung, "Protecting end-device from replay attack on lorawan," in *2018 20th International conference on advanced communication technology (ICACT)*, 2018.

[19] F. e. a. Hessel, "Chirpotle: a framework for practical lorawan security evaluation," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020.

[20] K.-H. e. a. Lai, "Revisiting time series outlier detection: Definitions and benchmarks," in *Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[21] D. e. a. Li, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International conference on artificial neural networks*, pp. 703–716, Springer, 2019.

[22] D. e. a. Park, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, 2018.

[23] C. Aggarwal, "Outlier analysis. data mining, 2015."

[24] B. e. a. Zong, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.

[25] M.-L. e. a. Shyu, "A novel anomaly detection scheme based on principal component classifier," tech. rep., Miami Univ Coral Gables Fl, 2003.

[26] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *International Conference on Knowledge Discovery in Data Mining*, 2005.

[27] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *European conference on principles of data mining and knowledge discovery*, pp. 15–27, Springer, 2002.