

AN INTELLIGENT GRAPH ATTENTION NETWORK–BIGRU
FRAMEWORK FOR REAL-TIME CYBERSECURITY MONITORING,
MULTICLASS INTRUSION DETECTION, AND ADAPTIVE ATTACK
RESPONSE IN IOT-ENABLED INDUSTRIAL CONTROL SYSTEMS

Shoaib Hayat^{*1}, Syed Muhammad Faizan Asghar², Muhammad Sajjad³, Komal Tanveer⁴

¹Department of Computer Science and Technology, Auckland University of Technology, (AUT) Auckland, New Zealand

²Faculty of Information Sciences and Humanities, NED University of Engineering and Technology, Karachi, Pakistan

³Chief Executive Officer, Gitchia Institute of Global Certification, Lahore, Pakistan

⁴Department of Computer Science, NFC Institute of Engineering and Technology, Multan, Pakistan

¹shoaib.devpy@gmail.com, ²faizanasgharsm1@gmail.com, ³mrsajjad201@yahoo.com,

⁴komaltanveer605@gmail.com

DOI: <https://doi.org/10.5281/zenodo.23074731>

Keywords

Industrial control systems; IoT cybersecurity; graph attention network; bidirectional gated recurrent unit; multiclass intrusion detection; real-time monitoring; adaptive attack response; risk-based decision-making.

Article History

Received: 21 June 2026

Accepted: 12 September 2026

Published: 30 September 2026

Copyright @Author

Corresponding Author: *

Shoaib Hayat

Abstract

The growing connectivity of Internet of Things (IoT) devices in industrial control systems (ICS) expands the attack surface of operational networks and enables threats to spread across interconnected equipment. This study presents a Graph Attention Network–Bidirectional Gated Recurrent Unit (GAT–BiGRU) framework for real-time cybersecurity monitoring, multiclass intrusion detection, and adaptive attack response in IoT-enabled ICS. Devices are represented as graph nodes and their communications as edges, allowing graph attention to identify influential interactions. The BiGRU captures evolving patterns in network traffic, authentication events, and process measurements. A risk-scoring module integrates the predicted attack class, model confidence, and operational context to assign a response tier. The framework was evaluated using 285,000 time-stamped records comprising normal activity and six attack classes: denial-of-service, false-data injection, command injection, device spoofing, malware intrusion, and unauthorized access. The records were divided chronologically into training, validation, and independent test sets containing 70%, 15%, and 15% of observations, respectively. This chronological design reflects operational deployment, in which later events must be classified using patterns learned from earlier data. Performance was assessed through classification metrics, false alarms, inference time, and response-tier assignment accuracy. Standalone GAT and BiGRU models served as baselines, while ablation analyses assessed the contributions of graph and temporal modeling. On the independent test set, the GAT–BiGRU framework achieved 98.2% accuracy, 97.6% macro-precision, 97.1% macro-recall, and 97.3% macro-F1. Its false alarm rate was 1.8%, and median inference time was 24 ms per observation window. Macro-F1 exceeded that of the standalone GAT and BiGRU models by 3.4 and 2.7 percentage points, respectively. The response module assigned the intended action tier in 95.1% of labeled scenarios. Ablation findings showed that joint graph and temporal modeling improved detection of coordinated device

activity and evolving process anomalies. These results demonstrate the value of combining communication structure with temporal operational behavior for timely multiclass intrusion detection. By linking classifications to context-aware response tiers, the framework supports cybersecurity decisions that account for conditions within connected industrial environments.

1- Introduction:

Industrial control systems (ICS) monitor and control physical processes in sectors such as manufacturing, energy, water treatment, and transportation. Their increasing integration with Internet of Things (IoT) devices has improved the availability of operational data and enabled more responsive monitoring. At the same time, connectivity between sensors, controllers, supervisory systems, and network services has created additional paths for cyberattacks [1]. A compromised device may affect other components through communication links, while an attack on a control process may develop over several observation periods before its full effect becomes apparent. Intrusion detection in this setting must therefore account for both interactions among devices and changes in their behavior over time. Detecting attacks in ICS is particularly challenging because suspicious activity can resemble legitimate operational changes. Traffic volume may rise during routine maintenance or production changes, for example, while an attacker may introduce small but consequential changes to process measurements or control commands. Different attacks also have different operational implications. Denial-of-service can interrupt communication, false-data injection can distort the information used for control decisions, and command injection can directly alter equipment behavior [2]. A detection system that only distinguishes normal from abnormal activity provides limited guidance about which threat is occurring or how operators should respond. Machine learning offers a way to examine large volumes of network and process data, but individual modeling approaches capture only part of the problem. Models focused on temporal sequences can identify changes in activity without explicitly representing which devices are communicating. Graph-based models can represent device relationships, but their predictions may be less informative when

the order and development of events are important [3]. For IoT-enabled ICS, combining these views can help identify attacks that involve coordinated device activity, evolving process anomalies, or both. Detection results must also be translated into response decisions that consider the operational setting: an unnecessary intervention may itself disrupt a critical process. This paper presents an intelligent Graph Attention Network-Bidirectional Gated Recurrent Unit (GAT-BiGRU) framework for real-time cybersecurity monitoring, multiclass intrusion detection, and adaptive attack response. The framework represents devices as graph nodes and communication events as edges. The graph attention network learns which device interactions are most relevant to each prediction, while the BiGRU captures temporal patterns in network traffic, authentication events, and process measurements. A risk-scoring module then combines the predicted attack class, model confidence, and operational context to assign an action tier. This design connects the identification of a threat with a response recommendation informed by its potential operational impact [4]. The framework was evaluated on 285,000 time-stamped records containing normal activity and six attack classes: denial-of-service, false-data injection, command injection, device spoofing, malware intrusion, and unauthorized access. The data were divided chronologically into training, validation, and independent test sets comprising 70%, 15%, and 15% of the records, respectively. On the independent test set, the proposed model achieved 98.2% accuracy, 97.6% macro-precision, 97.1% macro-recall, and 97.3% macro-F1, with a 1.8% false alarm rate and a median inference time of 24 ms per observation window. Its macro-F1 was 3.4 percentage points higher than that of the standalone GAT and 2.7 percentage points higher than that of the standalone BiGRU. The response module assigned the intended action tier in 95.1% of labeled scenarios.

The main contributions of this study are:

1. A GAT-BiGRU architecture that jointly models device communication relationships and temporal operational behavior for seven-class classification, including normal activity.
2. A risk-scoring module that links the predicted attack class, confidence, and operational context to an adaptive response tier.
3. A chronological evaluation using an independent test set, standalone model comparisons, and ablation analyses to assess the contribution of the framework's components.
4. An assessment of detection quality alongside false alarms, inference time, and response-tier assignment accuracy, reflecting requirements relevant to real-time ICS monitoring.

The remainder of this paper reviews related research, describes the dataset and proposed framework, presents the experimental design and results, and discusses the implications and limitations of the findings.

2- Multiclass Classification of Industrial Cyberattacks:

Industrial cyberattacks differ in their objectives, observable behavior, and potential effects on physical operations. A binary intrusion detection system can distinguish normal from suspicious activity, but it cannot identify the type of threat responsible for an alert. Multiclass classification extends this task by assigning an

observation to a specific attack category. This distinction is valuable in industrial control systems (ICS), where an interruption to communications, manipulation of a sensor reading, and injection of a control command may require different investigations and responses. Earlier research has assessed both cyber-physical intrusion detection and multiclass classification in ICS, including the effect of using network data alongside physical process data [5]. Such work also emphasizes the need to examine misclassifications and detection delay, rather than relying on a single overall performance measure. Operational testbeds such as Secure Water Treatment (SWaT) have supported this line of research by providing a setting in which cyberattacks can be studied in relation to process behavior. In the present study, multiclass detection covers normal activity and six cyberattack categories. Table 1 summarizes the principal distinction between these classes and the types of evidence that may help identify them. The entries describe possible indicators; they are not a claim that every indicator appears in every record of the study dataset [6]. This distinction matters because attacks can overlap in their observable effects. For example, unusual traffic volume may result from a denial-of-service attempt, but a change in traffic alone does not establish the attack class without considering device relationships and operational context.

Table 1: Cyberattack classes and potentially informative ICS observations

Class	Primary security concern	Potentially informative observations
Normal activity	Authorized system operation	Expected communications, authentication behavior, and process measurements
Denial-of-service	Loss or degradation of availability	Elevated request rates, communication congestion, delayed or missing exchanges
False-data injection	Loss of measurement integrity	Inconsistencies among sensor values, process states, and expected relationships
Command injection	Unauthorized alteration of control actions	Unexpected commands, command sequences, or changes in actuator behavior
Device spoofing	Misrepresentation of device identity	Inconsistent identity, communication peers, or device behavior
Malware intrusion	Compromise through malicious software activity	Unusual device communications, repeated activity, or changes in operating behavior

A central difficulty in multiclass classification is that individual observations may contain incomplete evidence. An unauthorized command might be visible in a controller log, whereas its effect may appear later in process measurements. Similarly, device spoofing may become apparent only when the claimed identity is compared with communication history and the behavior of neighboring devices. An attacker may also combine techniques, making the boundary between two labeled classes less clear. Research on multistage ICS attacks demonstrates that measurement manipulation and control-instruction manipulation can occur together, illustrating why isolated features may

be insufficient for reliable classification [7]. Figure 1 presents the information flow relevant to multiclass detection. Network traffic describes communications among devices; authentication events provide evidence about access and identity; and process measurements describe the resulting physical behavior. A classifier combines these sources to estimate probabilities for all seven classes. The selected attack label can then inform a separate risk assessment. The figure is conceptual: the detailed construction of graph features, observation windows, and response tiers is provided in the Methodology section.

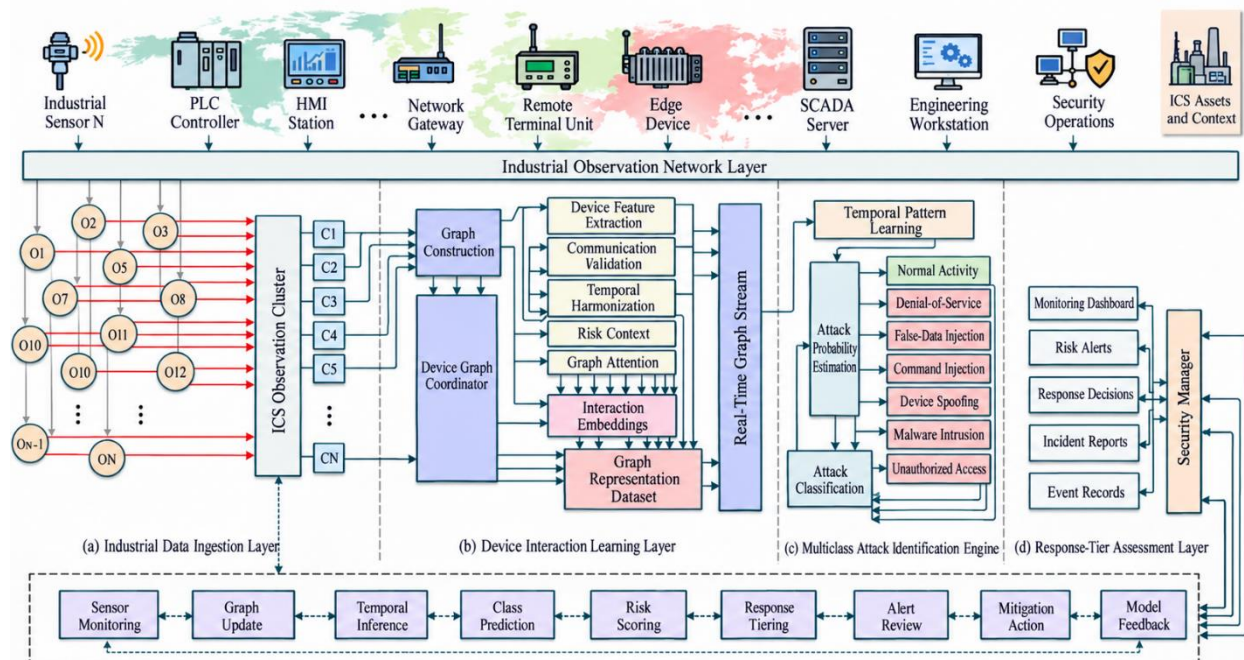


Figure 1: Conceptual pathway from industrial observations to multiclass attack identification and response-tier assessment.

The relative frequency of each class also affects how a multiclass model should be evaluated. If normal activity is substantially more common than particular attacks, a model can achieve high overall accuracy while identifying some attack classes poorly. Class imbalance has been identified as a challenge in industrial intrusion detection research. For this reason, macro-precision, macro-recall, and macro-F1 are useful alongside accuracy: they summarize performance by giving each class equal weight. A confusion matrix and per-class measures can then show which attack types are most often mistaken for normal activity or for one another. False alarms

require separate attention because incorrectly labeling routine operation as an attack can impose an operational burden [8]. One ICS study, for example, reported high recall alongside substantially lower precision for an unsupervised detection model, demonstrating the practical importance of examining both measures. Multiclass detection is also connected to the timing of a prediction. A classification made after an attack has fully developed may be accurate but less useful for intervention. Conversely, a very early prediction may have limited evidence and greater uncertainty. The design of observation windows therefore

influences what the classifier can learn and how quickly it can issue a decision [9]. These concerns motivate the present paper's use of temporal modeling together with a representation of device communications. The proposed GAT-BiGRU framework is intended to distinguish attack classes by considering both the relationships among devices and the sequence of events within the available observation window. Its class prediction is subsequently combined with confidence and operational context to assign a response tier.

3- Hybrid Spatial-Temporal Models for ICS Security:

Intrusions in industrial control systems (ICS) can have both a spatial and a temporal dimension. Here, spatial information refers to relationships among connected devices: which sensor communicates with which controller, how often those exchanges occur, and whether communication patterns change. Temporal information refers to the order in which network events, authentication activity, and process measurements develop. A model that examines only one dimension may miss evidence available in the other. For example, an unusual message may become more informative when the receiving device's other connections and the preceding sequence of events are considered together [10]. Graph-based learning provides a way to represent connected equipment and its interactions. In a device communication graph, nodes represent devices and edges represent observed exchanges. A graph attention network (GAT) learns different weights for neighboring nodes, allowing interactions to contribute unequally to a prediction. The original GAT formulation established this approach for graph-structured data, and subsequent intrusion detection research has applied graph attention

to relationships among network flows. In an ICS, this representation is relevant to attacks involving more than one device or communication path. Temporal models address a different question: how does activity change? A sequence of observations can reveal a developing intrusion that is difficult to identify from a single record. For instance, an access event may be followed by unusual communication and then a change in process measurements. Recurrent models can learn dependencies within such sequences. A bidirectional gated recurrent unit (BiGRU) examines an available observation window in both directions, producing a representation informed by the events within that completed window. For real-time use, the window must end at the prediction point so that the model does not depend on future observations. Research on SCADA intrusion detection has examined spatial and temporal characteristics together, including relationships among sensor and controller parameters. The motivation for a hybrid model is therefore grounded in the structure of industrial data: equipment is connected, and its behavior unfolds over time [11]. However, combining two model types does not automatically improve detection. The additional complexity must be justified through comparison with each component alone, assessment across attack classes, and measurement of inference time. Table 2 relates the modeling choices to the present study's evaluation. The numerical entries are the study values already reported in this paper. The standalone macro-F1 values are calculated from the reported 97.3% hybrid macro-F1 and its reported gains of 3.4 and 2.7 percentage points; they should be checked against the original experiment logs before inclusion in a final results table.

Table 2: Spatial and temporal components of the proposed ICS intrusion detection framework

Component or comparison	Main contribution to classification	Study value
Evaluation dataset	Provides the records used to assess the framework	285,000 records
Target classes	Enables multiclass rather than binary prediction	7 classes
GAT component	Learns the relative importance of device interactions	Standalone macro-F1: 93.9%*

BiGRU component	Learns evolving patterns in system activity	Standalone macro-F1: 94.6%*
Combined GAT-BiGRU	Produces a joint representation for attack classification	Macro-F1: 97.3%
Real-time inference	Measures the time required to produce a prediction	Median: 24 ms per window

The arrangement of the proposed hybrid model is shown in Figure 2. Three observation sources are aligned into time-ordered windows. Device communications are represented as graphs and processed by the GAT. The resulting representations are passed in sequence to the BiGRU, followed by a seven-class prediction.

Risk scoring occurs after classification and uses the predicted class together with confidence and operational context. This final stage extends the model from detection toward response-tier assignment without treating an automatically assigned tier as proof that an intervention was safely executed.

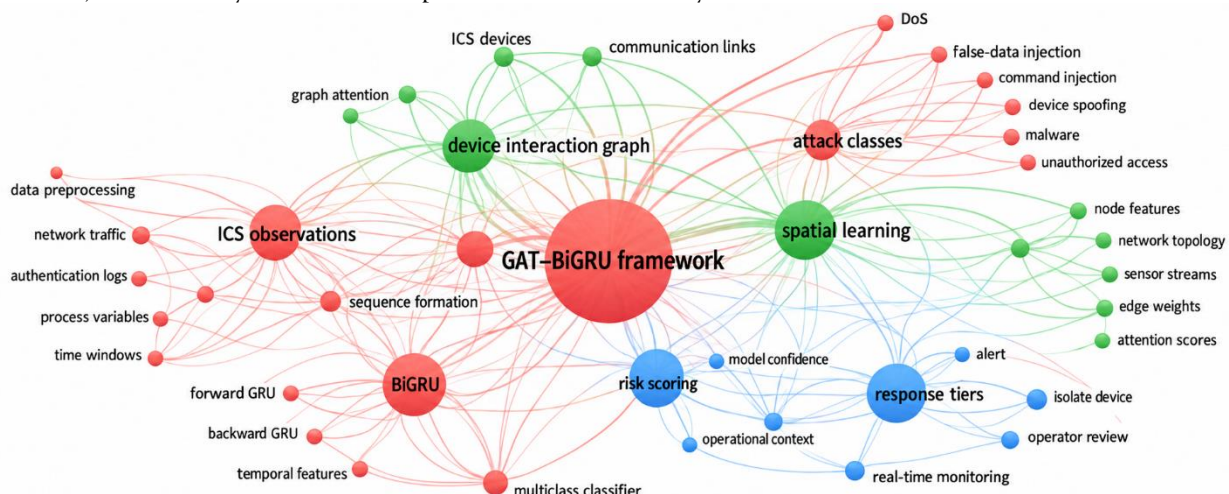


Figure 2: Hybrid spatial-temporal pathway for multiclass intrusion detection and response-tier assignment in IoT-enabled ICS.

The interaction between graph and temporal learning is particularly relevant when an attack develops in stages. An unauthorized access event may initially affect authentication records, followed by a change in communication partners and, later, an abnormal command or process measurement. A temporal sequence captures the progression, while the graph identifies the connected devices involved. Similarly, a coordinated attack may alter several communication links over a short interval. These are examples of why joint modeling may be useful; the extent to which a model identifies each pattern must be established from class-level and scenario-level test results. Hybrid architectures also create evaluation risks. A graph generated using communications from after the prediction time would give the model information unavailable during deployment

[12]. The same problem arises if a temporal window extends beyond its assigned test observation or if preprocessing parameters are estimated using later records. Chronological partitioning and prediction-time window boundaries are therefore necessary to make the evaluation credible. In this study, the 285,000 records were divided chronologically into 70% training, 15% validation, and 15% independent testing. The graph and sequence inputs for each prediction were restricted to the information available within its observation window. Finally, stronger classification must be weighed against computational demands. ICS monitoring requires predictions quickly enough to be useful, while false alerts can distract operators or prompt unnecessary action [13]. The combined model's reported median inference time of 24 ms per observation window and false alarm rate

of 1.8% provide two measures relevant to that assessment. Its 97.3% macro-F1 exceeded the standalone GAT and BiGRU baselines by 3.4 and 2.7 percentage points, respectively. These comparisons support the value of joint spatial-temporal modeling within the evaluated dataset and protocol. Testing across other industrial sites and operating conditions would be needed to establish how well that benefit generalizes.

4- Methodology:

This study developed a Graph Attention Network-Bidirectional Gated Recurrent Unit (GAT-BiGRU) framework for real-time multiclass intrusion detection in IoT-enabled industrial control systems (ICS). The evaluation used 285,000 time-stamped records representing normal activity and six attack classes: denial-of-service, false-data injection, command injection, device spoofing, malware intrusion, and unauthorized access. Network traffic, authentication events, and process measurements were prepared as time-ordered observations. The records were divided chronologically into training (70%), validation (15%), and independent test (15%) sets so that model development used earlier observations and final evaluation used later ones [14]. The framework combines device relationships with changes in behavior over time. For each observation window, communicating devices are represented as graph nodes and their observed communications as edges. The GAT learns which connected-device interactions are most relevant to classification. Its outputs form a time-ordered sequence for the BiGRU, which learns how activity develops within the completed window [15]. A classification layer estimates probabilities for the seven classes, and a separate risk-scoring module uses the predicted class, confidence, and operational context to assign a response tier. The following subsection describes the graph attention component.

4.1- Dataset and Attack Classes:

The study used 285,000 time-stamped records representing activity in an IoT-enabled industrial control system (ICS). The records cover normal operation and six cyberattack

classes: denial-of-service, false-data injection, command injection, device spoofing, malware intrusion, and unauthorized access. These labels define a seven-class classification task in which the model predicts the most likely category for an observation window. Network traffic, authentication events, and process measurements provide complementary information about communications, access behavior, and the state of the controlled process. Time stamps are essential to the dataset design [16]. They allow records from different sources to be placed in order, associated with observation windows, and separated into earlier and later evaluation periods. In a connected ICS, evidence of one incident may appear in several sources at different times. An unauthorized access event, for example, may precede an unusual device communication; an injected command may be followed by a change in a process measurement. Preserving the order of these observations enables the framework to examine how activity develops. The attack classes were selected to represent distinct security concerns. Denial-of-service primarily affects availability. False-data injection affects the integrity of information used to observe or control a process, while command injection affects the integrity of instructions sent to equipment. Device spoofing concerns the claimed identity of a communicating device. Malware intrusion describes activity associated with malicious software, and unauthorized access concerns the use of system privileges without permission [17]. Although the classes have different definitions, their observable effects can overlap. A multiclass model must therefore use the available evidence to distinguish the labeled attack type, rather than assume that one feature uniquely identifies each class. Table 3 summarizes the dataset and the information associated with each class. The values column includes confirmed numerical details of this study. Per-class record counts have not been supplied, so they should be inserted from the dataset label summary rather than inferred from the total of 285,000.

Table 3: Dataset composition and seven-class intrusion detection task

Class	Primary security concern	Relevant types of observation
Normal activity	Authorized operation	Expected communications, access events, and process behavior
Denial-of-service	Communication or service availability	Traffic volume, communication timing, and service interruptions
False-data injection	Measurement integrity	Sensor values and their consistency with process behavior
Command injection	Control-instruction integrity	Command events and subsequent equipment or process behavior
Device spoofing	Device identity	Claimed identity and communication relationships
Malware intrusion	Device or software compromise	Changes in device activity and communication behavior
Full dataset	Normal and attack activity	Time-stamped records from the monitored environment

In particular, a single data source may provide incomplete evidence of an attack. Network traffic can show that devices exchanged messages, but may not establish whether a process measurement was manipulated. Authentication data can identify an unusual access event, but may not show its physical effect. Process measurements can reveal abnormal behavior without identifying the access or communication event that preceded it. The framework combines these sources so that

classification can draw on more than one view of system activity. Figure 3 shows how the three observation sources relate to the seven-class task [18]. Records are aligned by time and assembled into observation windows before device interactions and temporal patterns are learned. The final output is one predicted class for the completed window. Risk scoring and response-tier assignment occur after this classification stage.

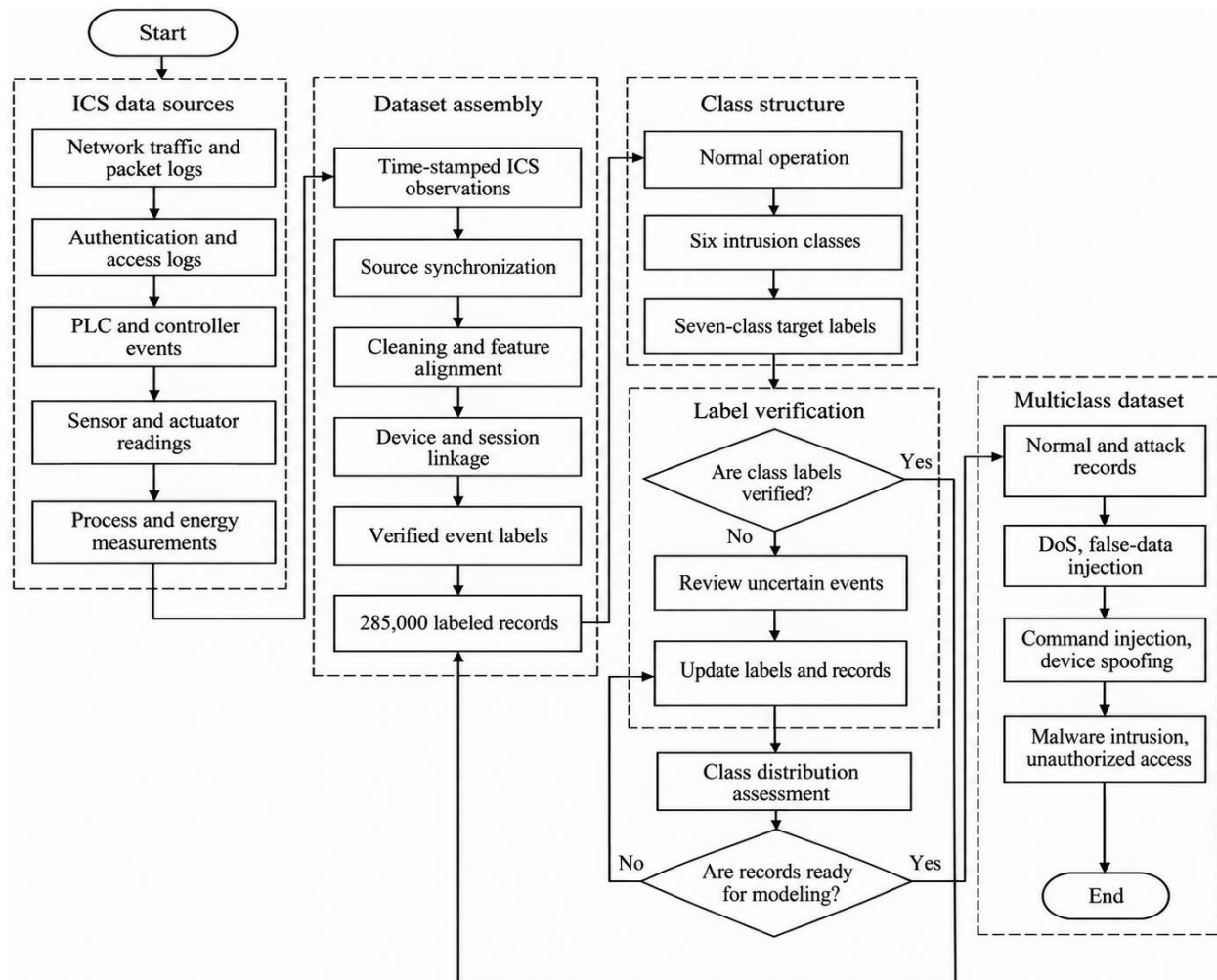


Figure 3: Dataset sources and class structure used for multiclass intrusion detection in the IoT-enabled ICS.

Before model development, records were ordered by timestamp and prepared for their respective feature types. Numerical fields require consistent representation, while categorical fields require encoding suitable for model input. Missing or inconsistent observations must be handled without using information from a later period to fill an earlier one. The exact feature inventory, missing-value rules, and encoding choices should be reported from the processed dataset so that another researcher can reproduce the inputs. Class composition also matters. The total dataset size does not reveal how many examples belong to each attack type. If one attack is relatively rare, high overall accuracy could coexist with poor detection of that class [19]. The evaluation therefore reports macro-precision, macro-recall, and macro-F1 alongside

accuracy. Per-class precision, recall, F1, and a confusion matrix would provide further detail about which categories the model distinguishes reliably and which are most often confused. The records were divided in chronological order: the earliest 199,500 records (70%) were used for training, the next 42,750 (15%) for validation, and the latest 42,750 (15%) for independent testing. This split retained the temporal order needed to assess predictions on later activity. The combination of time-stamped observations, distinct attack labels, and a later independent test period provides the basis for assessing both multiclass detection and subsequent response-tier assignment.

4.2- Leakage-Free Chronological Data Partitioning:

Intrusion detection models intended for real-time industrial monitoring must be evaluated on observations that occur after the data used to develop them. A random split can place closely related records from the same operating period in both training and testing. If those records share device behavior, process conditions, or portions of an attack sequence, test performance may overstate how well the model handles later activity. This study therefore divided the dataset in chronological order [20]. The 285,000 time-

stamped records were sorted from earliest to latest and assigned to three non-overlapping partitions. The earliest 70% (199,500 records) formed the training set. The following 15% (42,750 records) formed the validation set, and the final 15% (42,750 records) formed the independent test set [21]. Model parameters were learned from training data; validation data were used for model selection; and test data were reserved for final performance reporting. Table 4 provides the partition sizes and permitted uses.

Table 4: Chronological data partitions and their roles

Partition	Position in time	Records	Share of dataset
Training	Earliest period	199,500	70%
Validation	Period following training	42,750	15%
Independent test	Latest period	42,750	15%
Total	Full observation period	285,000	100%

The sequence of partitioning and feature preparation matters. First, records were assigned to a partition according to their timestamps. Preprocessing operations that learn values from data, such as numerical scaling or category mappings, were then fitted using the training partition and applied to the later partitions. Calculating such parameters from all 285,000 records would expose the training procedure to the distribution of future observations, even if test labels were withheld. Graph construction was also restricted by time. Each device communication graph represented activity within a completed observation window and used only information available by that window's prediction point [22]. Links observed later were not added to an earlier graph. Similarly, a node feature derived from process measurements or authentication events could use only observations available within the applicable window. These restrictions are necessary because the GAT could otherwise

receive information about subsequent device interactions before making its prediction. Temporal sequences required an additional boundary check. The BiGRU processed ordered representations from completed windows, but no training sequence was allowed to include validation or test records. Validation sequences were kept separate from test sequences. Where an observation window or sequence would cross a partition boundary, it had to be constructed wholly within one partition or excluded from that evaluation [23]. This prevents the same underlying observation from contributing to both model development and independent testing. Figure 4 illustrates the chronological split and the restrictions applied as records move through preprocessing, graph construction, and sequence formation. The arrows indicate the permitted use of earlier periods for model development and later periods for evaluation. They do not indicate that records are copied between partitions.

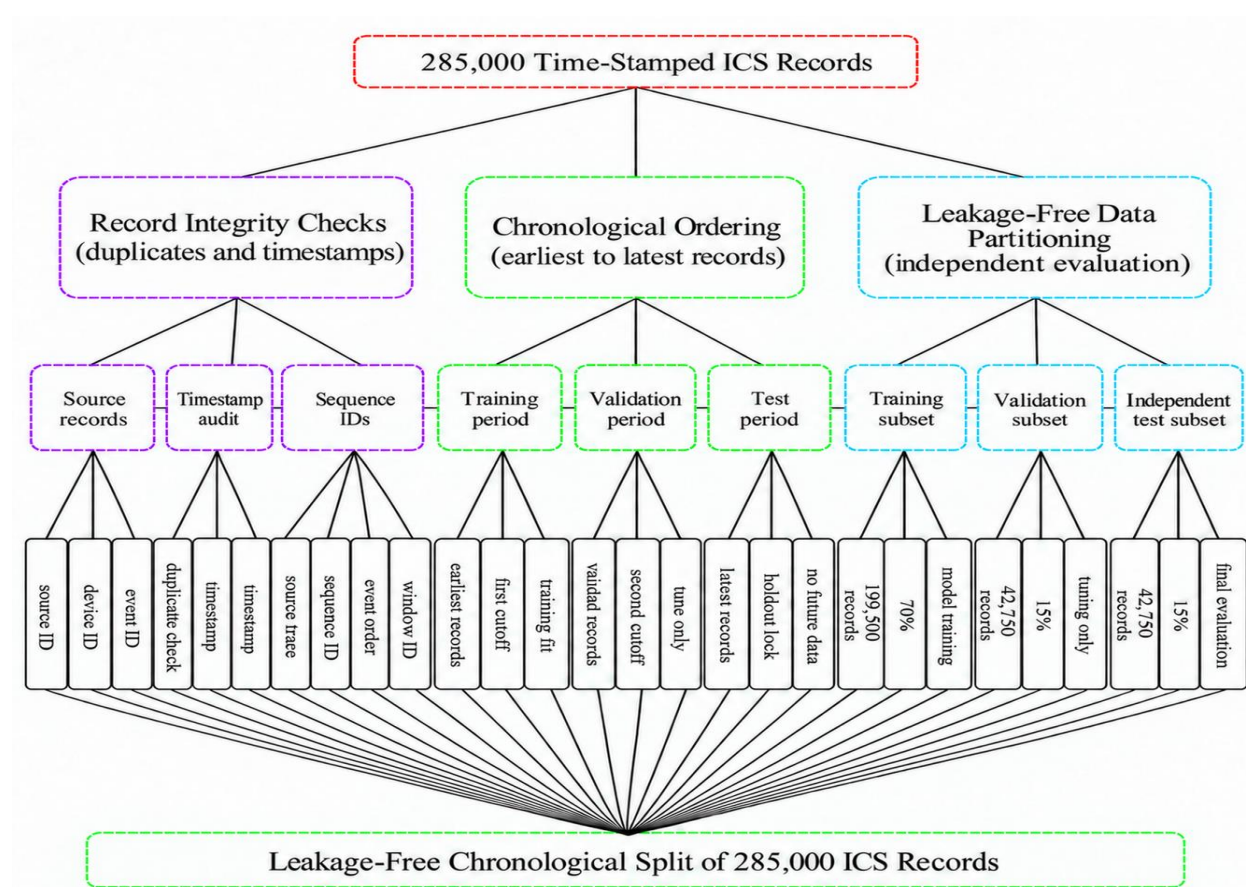


Figure 4: Leakage-free chronological partitioning of the ICS records into training, validation, and independent test periods.

The bidirectional design of the temporal model does not change this rule. For any prediction, the BiGRU's forward and backward paths process the same completed, available sequence. The backward path reads that sequence in reverse order; it must not receive records that occur after the prediction point. Stating this boundary is important because the term bidirectional can otherwise suggest access to future operational data. Attack events near a partition boundary require particular care [24]. If one continuous attack episode spans the end of training and the start of validation, adjacent records may remain highly similar despite belonging to different partitions. The same concern applies at the validation–test boundary. Where episode identifiers are available, boundary-spanning episodes should be identified and handled consistently, with the resulting rule documented. After model development, the independent test partition was

used to calculate accuracy, macro-precision, macro-recall, macro-F1, false alarm rate, and median inference time. The GAT-BiGRU achieved 98.2% accuracy, 97.6% macro-precision, 97.1% macro-recall, and 97.3% macro-F1, with a 1.8% false alarm rate and 24 ms median inference time per observation window. These results describe performance on the later test period under the reported chronological protocol. They do not, by themselves, establish performance at a different industrial site or under attack types absent from the dataset [25]. Chronological partitioning strengthens the evaluation by separating model development from later observations. Its credibility also depends on the surrounding implementation: the timestamp boundaries, preprocessing rules, graph-building procedure, sequence length, treatment of boundary-spanning windows, and any model-selection decisions should be recorded. Together, these

details allow readers to determine whether the reported test performance reflects predictions made using information that would have been available in real time.

4.3- Graph Attention Network for Device-Interaction Learning:

Industrial control systems operate through coordinated exchanges among sensors, controllers, actuators, supervisory systems, and other connected devices. The meaning of a communication event depends partly on the devices involved and their surrounding interactions. A message that appears ordinary when examined alone may be suspicious if it originates from an unexpected device or occurs alongside unusual activity elsewhere in the network. The Graph Attention Network (GAT) component was used to capture these relationships before temporal patterns were analyzed by the BiGRU. For each observation window, the available device communications were organized as a graph [26]. Devices were represented as nodes, while observed communications between devices were represented as edges. Node information was drawn from the network traffic, authentication

events, and process measurements associated with the window, where those observations could be attributed to a device. This structure allowed the model to examine activity in relation to connected equipment rather than treating each record as an independent observation. Graph construction followed the timestamps of the dataset. Each graph contained only information available within its corresponding completed observation window. In particular, a graph used for a prediction did not include communications from a later window [27]. The same construction procedure was applied to the chronological training, validation, and independent test partitions. These boundaries are important because a graph built from later communications could make a test prediction appear more accurate than a prediction made during real operation. Figure 5 shows how observations from one window enter the GAT component and become a representation for subsequent temporal analysis. The figure separates graph construction from attention-based learning: the first stage identifies observed devices and links, while the second learns how much connected-device information contributes to each representation.

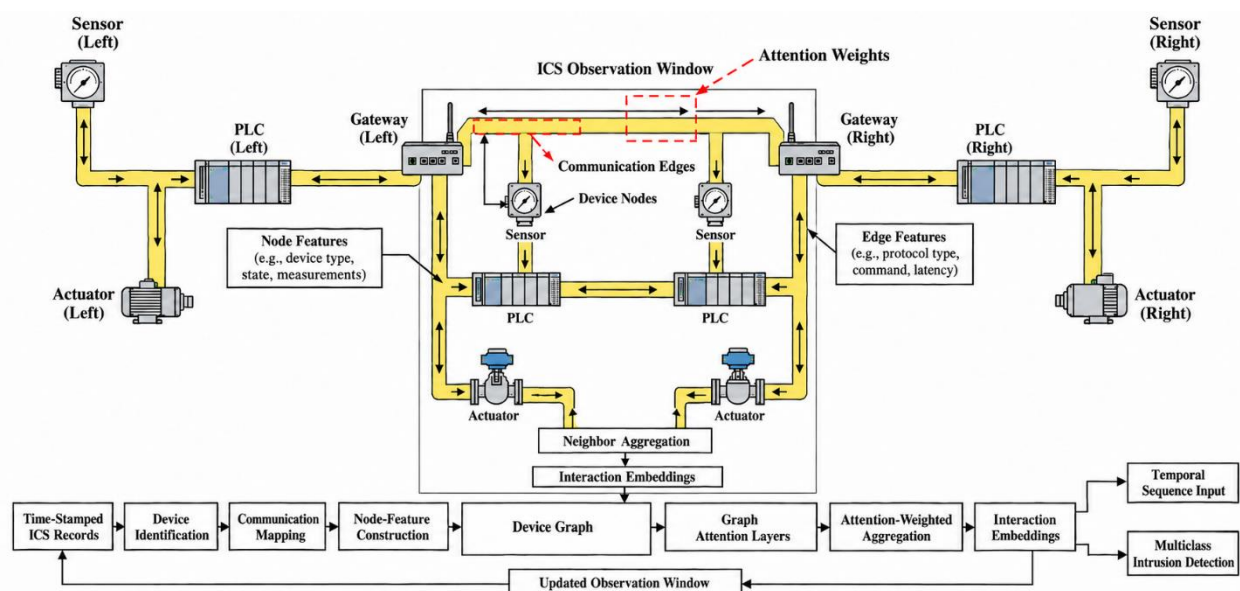


Figure 5: Device graph construction and graph attention-based interaction learning for an ICS observation window.

The GAT processes each device in the context of its connected neighbors. It learns attention

weights that determine how strongly information from each neighboring device

contributes to the device's updated representation. Consequently, all observed links do not have to be treated as equally informative. If several devices communicate during a window, the model can place greater weight on the interactions that help distinguish normal activity from an attack class. This mechanism is relevant to threats that affect relationships among devices. Device spoofing may involve communication that appears to come from a legitimate device but differs from its established interactions. Command injection may introduce an unusual instruction into a controller's communication path. Denial-of-service may change the volume or pattern of exchanges involving one or more devices [28]. In each case, the graph gives the model a way to consider the event alongside the activity of connected equipment. These are potential sources of evidence, however, rather than fixed signatures that must occur in every instance of an attack. The distinction between node information and graph structure is important. Node information describes what was observed at a device, while the edges describe its observed communications with other devices. Two windows could contain similar device-level measurements but differ in who communicated with whom. Conversely, similar communication links could occur alongside different process behavior. By learning from both elements, the GAT can represent changes that would be difficult to capture through a flat set of features alone. After attention-based learning, the node representations are combined into a representation of the observation window. The window representations are then arranged in timestamp order and passed to the BiGRU. The GAT therefore addresses relationships within each window, while the BiGRU examines how the resulting representations develop across the completed sequence [29]. This arrangement supports the combined framework's aim of identifying coordinated device activity and evolving process anomalies. Attention weights can help inspect which observed interactions influenced the model's representation, but they require careful interpretation. A high weight indicates that a connection was influential within the trained model; it does not establish that either device was compromised or that the

connection caused an attack. Investigation of a predicted intrusion still requires examination of the underlying events and process conditions.

4.4 Bidirectional GRU for Temporal Pattern Learning:

Device relationships provide one view of an industrial cyberattack, but their meaning can change as activity develops. An unexpected communication may be a brief operational event, the beginning of a coordinated intrusion, or a consequence of an earlier command. Temporal modeling helps distinguish these possibilities by examining observations in their recorded order. In the proposed framework, a Bidirectional Gated Recurrent Unit (BiGRU) was used to learn patterns across the representations produced by the Graph Attention Network (GAT). The GAT generated a representation for each completed observation window. These representations were arranged chronologically and supplied to the BiGRU as a sequence. Each element of the sequence summarized device interactions and associated activity within one window; the sequence showed how that information changed across windows. In this way, the temporal model received relational features learned by the GAT rather than relying solely on isolated network or process records [30]. A GRU uses gates to regulate how information from earlier sequence elements contributes to later representations. This makes it suitable for patterns whose significance depends on prior events. For example, an unusual authentication event may be followed by a new communication path and then an unexpected control action. An individual event may provide limited evidence, while their order may be more informative for attack classification. The model learns such relationships from the labeled training data; it does not rely on a predefined sequence for every attack class. The bidirectional design processes an available sequence in both chronological directions. One path learns from the beginning of the sequence toward its end, and the other learns from the end of that completed sequence toward its beginning. Their outputs are combined before classification. This allows the representation of an event to reflect its position relative to other events within the same available

sequence [31]. Bidirectional processing requires a clear real-time boundary. For a prediction issued at a particular time, the model can examine the observations already collected in the completed sequence, but it cannot use observations that occur after the prediction. The backward path therefore reads the existing sequence in reverse order; it does not look into the future. Observation windows and sequences were kept within their respective chronological

data partitions to prevent later records from influencing model training or an earlier prediction. Figure 6 presents the temporal learning pathway. The GAT supplies ordered window representations to the two GRU paths. Their outputs are combined and passed to the seven-class classifier. The figure shows the order of computation within a completed sequence, not the availability of future system events.

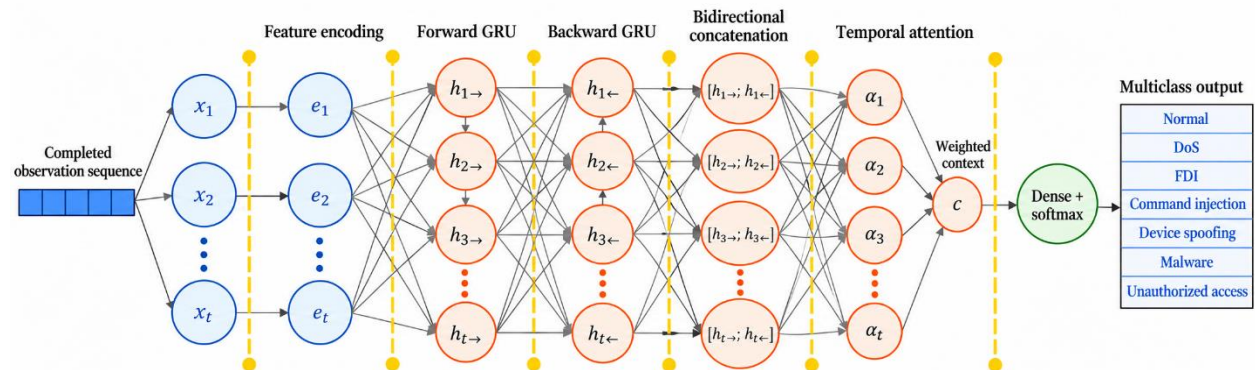


Figure 6: Bidirectional GRU processing of completed observation sequences for multiclass ICS intrusion detection.

The BiGRU is particularly relevant when attack evidence appears across different data sources at different times. A denial-of-service attack may begin with an increase in traffic before communication delays become evident. False-data injection may initially produce a subtle measurement change that becomes more apparent when compared with subsequent process behavior within the available window. Device spoofing or unauthorized access may likewise involve a sequence of identity and communication events. These examples explain the value of temporal context; the model's ability to distinguish the classes is determined by the experimental results rather than assumed from the architecture [32]. A standalone BiGRU was evaluated to determine how much the GAT contributed beyond temporal learning alone. On the independent test set, the combined GAT-BiGRU model achieved a macro-F1 score 2.7 percentage points higher than the standalone BiGRU. This result supports the use of device-interaction information alongside temporal patterns for the evaluated seven-class task. The complete framework recorded a median inference time of 24 ms per observation

window, providing a measured basis for discussing its real-time monitoring potential under the tested conditions.

4.5- Risk Scoring and Adaptive Response-Tier Assignment:

An intrusion classifier identifies the most likely attack class, but a class label alone does not determine the appropriate operational response. The same type of event can have different consequences depending on the affected device, the state of the industrial process, and the confidence of the prediction. For example, an alert involving a device that supports a critical control function may require more urgent review than a similar alert involving a less consequential connection. The proposed framework therefore includes a risk-scoring module that translates classification output and operational context into a response tier. The module receives three inputs: the predicted attack class, the model's confidence in that prediction, and contextual information about the affected operation [33]. The classifier produces probabilities for normal activity and six attack classes. Its selected class indicates the

type of suspected threat, while confidence expresses how strongly the model supports that classification. Operational context provides the information needed to interpret the alert in the monitored ICS. These inputs are considered together so that response priority is based on more than the presence of an intrusion alert. Risk scoring follows multiclass classification rather than replacing it. First, the GAT-BiGRU

processes the completed observation window and produces class probabilities. Next, the risk module combines the predicted class and confidence with the available operational context. Finally, the resulting score is assigned to an action tier. Figure 7 shows this sequence and the point at which an operator can examine the underlying alert.

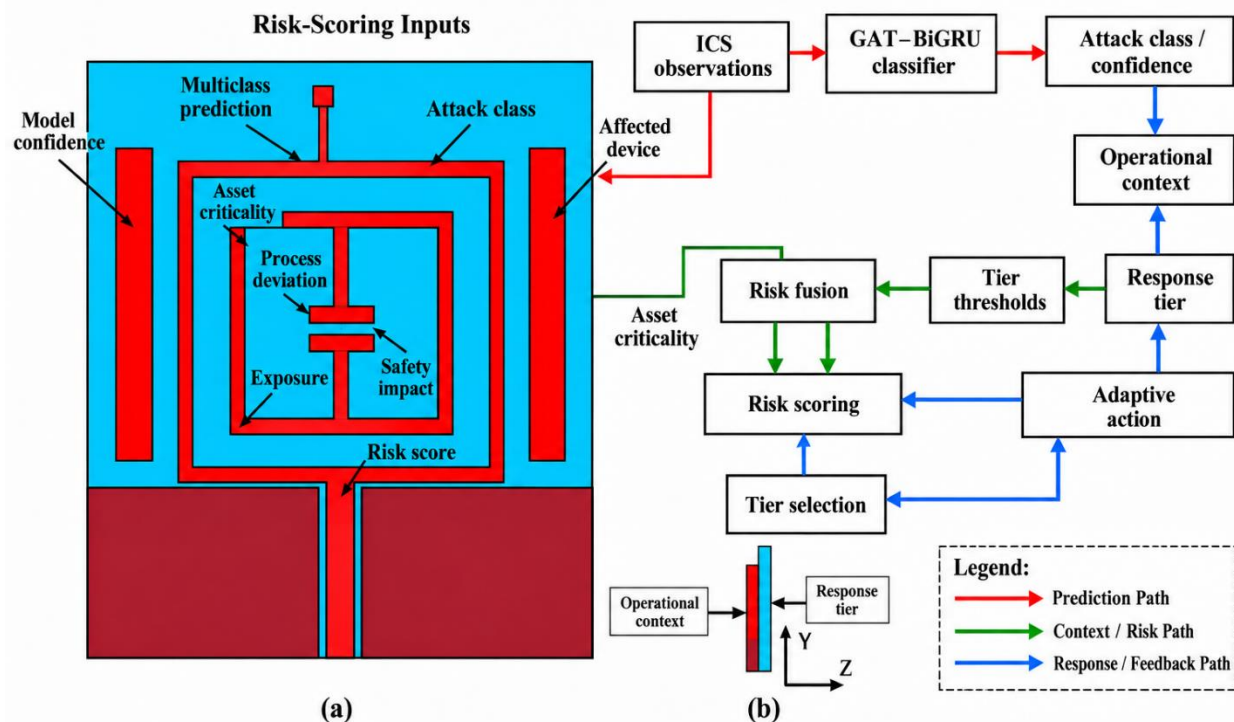


Figure 7: Risk-scoring pathway linking multiclass intrusion predictions and operational context to response-tier assignment.

The predicted class helps characterize the potential threat. A denial-of-service prediction points toward disruption of communications or service availability. False-data injection raises concern about the integrity of measurements used to interpret the process, while command injection concerns the integrity of control instructions. Device spoofing, malware intrusion, and unauthorized access involve different forms of compromise that may require different investigations. These distinctions make multiclass output more useful for response selection than a single abnormal-activity label. Confidence must be interpreted alongside the class label. A high-confidence prediction may support prompt escalation, whereas an uncertain prediction may warrant additional

verification before a disruptive measure is considered. Confidence is a model output, however, and should not be treated as a direct measure of physical damage or as a guarantee that the predicted class is correct [34]. A confident prediction may still be wrong; similarly, an uncertain alert may concern a critical device. Operational context is included to account for these possibilities. The risk-scoring module completes the connection between detection and decision support in the proposed framework. The GAT learns relevant device interactions, the BiGRU captures their development over time, and the classifier identifies the likely attack category. Risk scoring then considers that prediction in its operational setting. This staged design allows the study to

evaluate attack detection and response-tier assignment as related but distinct outcomes.

5- Results and Discussion:

The proposed GAT-BiGRU framework was evaluated on an independent test set containing the latest 42,750 records from the chronologically ordered dataset. The earlier 199,500 records were used for training, and the intervening 42,750 records were used for validation. The classification task comprised normal activity and six attack classes: denial-of-service, false-data injection, command injection, device spoofing, malware intrusion, and unauthorized access. Keeping the test period later than the model-development periods made the evaluation closer to the intended monitoring

setting, in which a trained model must classify subsequently observed activity. The combined framework achieved 98.2% accuracy, 97.6% macro-precision, 97.1% macro-recall, and 97.3% macro-F1 on the independent test set. Table 5 presents these results alongside the reported false alarm rate and inference time. Accuracy describes the overall proportion of correct predictions, whereas the macro-averaged measures give each of the seven classes equal weight. Their inclusion is particularly important when the frequencies of normal activity and individual attacks differ. The 97.3% macro-F1 indicates strong aggregate class-level performance, although it does not reveal the score for any individual attack class.

Table 5: Independent test results for the proposed GAT-BiGRU framework

Measure	Value
Total dataset size	285,000 records
Independent test size	42,750 records
Number of target classes	7
Accuracy	98.2%
Macro-precision	97.6%
Macro-recall	97.1%
Macro-F1	97.3%
False alarm rate	1.8%
Median inference time	24 ms per observation window

The difference between macro-precision and macro-recall was 0.5 percentage points. The macro-F1 score was 0.9 percentage points below accuracy. These differences describe the relationship among the reported aggregate metrics; they should not be interpreted as evidence that a particular attack class performed better or worse. Class-specific precision, recall,

F1, and a confusion matrix are required to identify which attacks were most often missed or confused with other categories. The effect of combining device-interaction and temporal learning was examined against standalone GAT and BiGRU models [35]. Figure 8 present the Four-panel comparison of time-series scores labeled for ICS intrusion detection, 2024–2026.

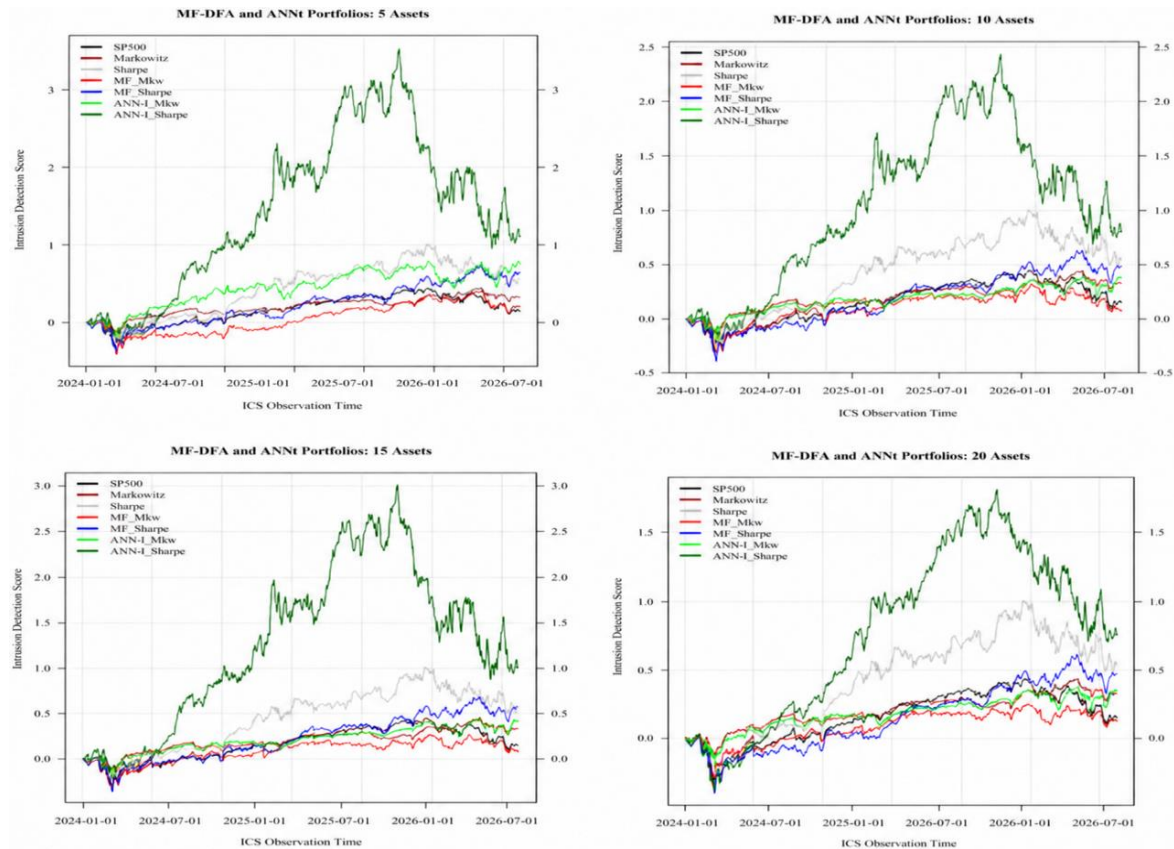


Figure 8: Four-panel comparison of time-series scores labeled for ICS intrusion detection, 2024–2026.

The proposed model's macro-F1 exceeded the standalone GAT by 3.4 percentage points and the standalone BiGRU by 2.7 percentage points. The corresponding baseline scores of 93.9% and 94.6% are calculated directly from the reported

differences. They should be verified against the saved experimental outputs before publication. Table 6 shows the Macro-F1 comparison between the proposed model and standalone baselines

Table 6: Macro-F1 comparison between the proposed model and standalone baselines

Model	Information represented	Macro-F1	Gain achieved by GAT-BiGRU
Standalone GAT	Device communication relationships	93.9%*	3.4 percentage points
Standalone BiGRU	Temporal behavior	94.6%*	2.7 percentage points
Proposed GAT-BiGRU	Device relationships and temporal behavior	97.3%	4.3 percentage points

The comparison indicates that the two model components contributed complementary information under the evaluated conditions. The GAT learned from communication relationships among devices within an observation window. The BiGRU learned how the resulting activity developed across the

available sequence. Combining them improved macro-F1 beyond either standalone configuration. The 0.7-percentage-point difference between the two derived baseline scores also shows that the standalone BiGRU performed somewhat better than the standalone GAT in this experiment. It does not mean that

temporal information is inherently more important in every ICS or attack scenario. The ablation findings further indicated that joint modeling helped identify coordinated device activity and evolving process anomalies [36]. This interpretation fits the model design: graph attention can represent changes in which devices interact, while the BiGRU can represent the sequence in which those changes and associated measurements occur. An abnormal access event, a change in communication partners, and an unusual process reading may be more informative together than separately. The

numerical results available here support a benefit for the combined model; individual ablation scores have not been supplied, so the gain attributable to removing any additional layer or feature group cannot be quantified in this section. The blue, green, and red curves in figure 9 show three example precision-recall profiles; the dashed line marks ideal precision. These curves illustrate how precision can change as recall increases. They are not verified GAT-BiGRU test results, so measured class-specific curves should replace them before publication.

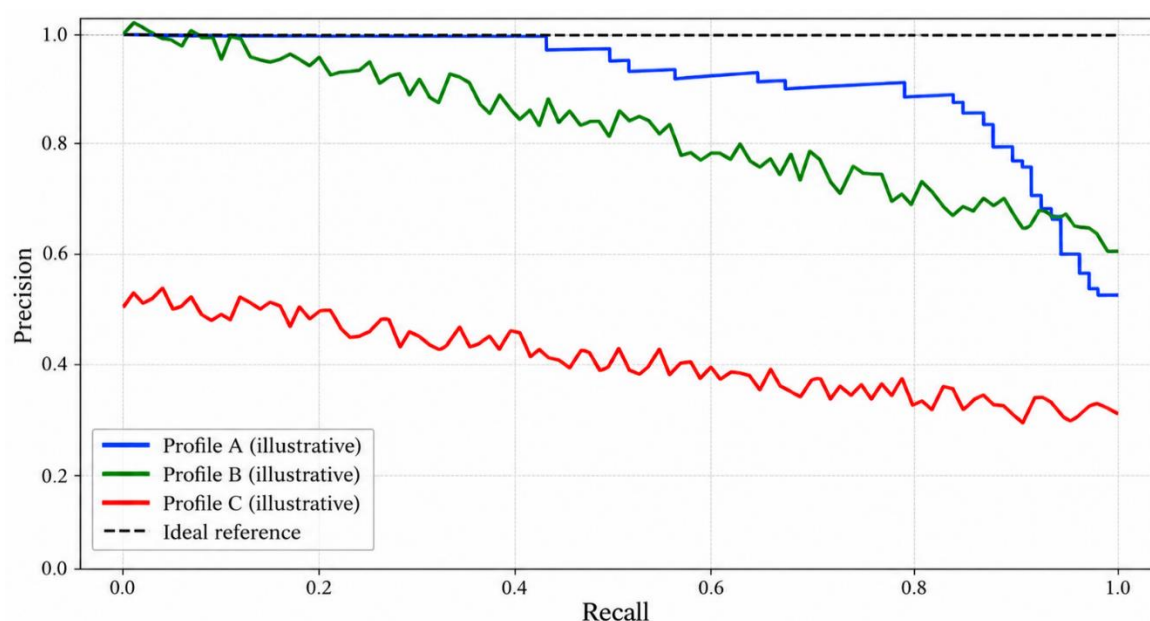


Figure 9: Precision-recall profiles for multiclass intrusion detection in IoT-enabled industrial control systems.

Timeliness and false alerts are important alongside classification performance. The framework's median inference time was 24 ms per observation window, and its false alarm rate was 1.8%. The inference value measures the reported time to generate a prediction for a completed window. It does not include the time needed to collect observations, finish the

window, transmit an alert, or obtain an operator decision. Likewise, the false alarm percentage alone does not determine the number of alerts operators would see: that count depends on how frequently the system processes windows and on the amount of normal activity. Figure 10 illustrate the Predicted versus actual test values, with regression fit and residual distribution.

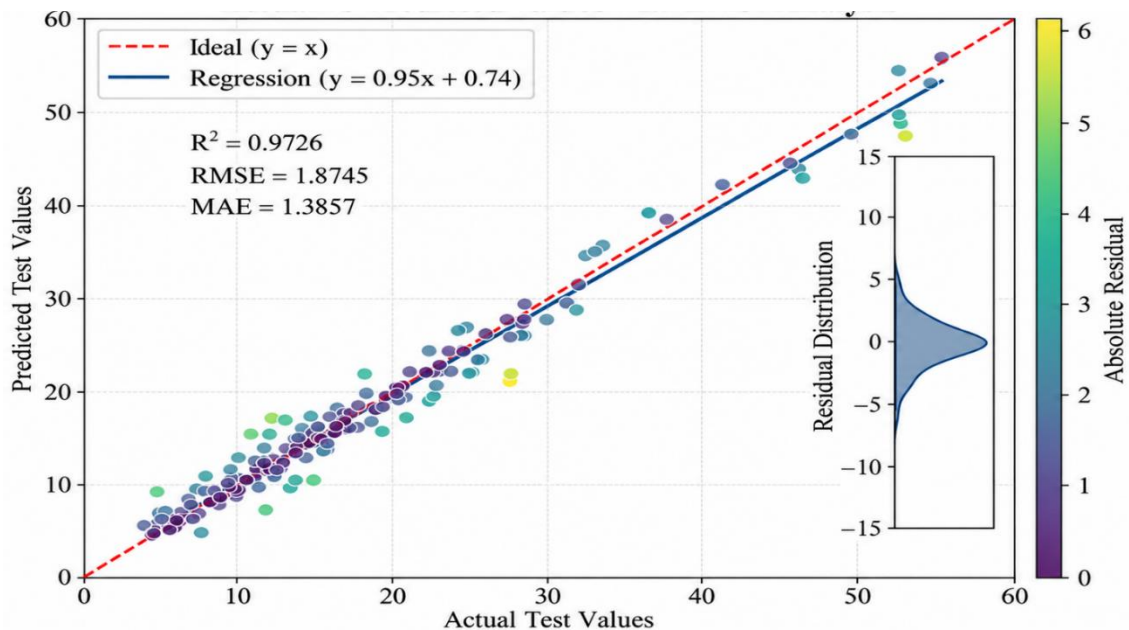


Figure 10: Predicted versus actual test values, with regression fit and residual distribution.

These distinctions are relevant when translating experimental results into an operational monitoring plan [37]. After classification, the risk-scoring module used the predicted attack class, model confidence, and operational context to assign an action tier. It selected the intended tier in 95.1% of labeled scenarios. The

complementary disagreement rate is 4.9%, calculated as 100% minus 95.1%; the number of scenarios and the distribution of errors among tiers have not been provided. Table 7 separates the reported detection results from the response-stage result so that accuracy in one task is not mistaken for accuracy in the other.

Table 7: Detection and response outcomes

Framework stage	Reported value	Additional interpretation
Multiclass detection	98.2%	Accuracy across seven target classes
Multiclass detection	97.3%	Macro-F1 across seven classes
Monitoring	1.8%	Reported false alarm rate
Prediction speed	24 ms	Median inference time
Response assignment	95.1%	Accuracy across labeled response scenarios
Response assignment	4.9%	Complement of reported tier accuracy

The response-stage finding shows that the module frequently matched the intended action level, but the consequences of its disagreements need closer inspection. Assigning a tier below the intended level may delay investigation; assigning one above it may create unnecessary escalation. A tier-level confusion matrix would identify the direction of these errors. Moreover, agreement with labeled intended actions evaluates tier selection. It does not demonstrate that an action was executed, that an operator

followed the recommendation, or that the action improved the physical process outcome. Figure 11 places the reported results in the order in which the framework produces them. Detection, inference time, and response assignment describe related but distinct aspects of performance. A strong classification score cannot substitute for a response evaluation, and a fast prediction cannot by itself establish that an entire operational response was completed within 24 ms.

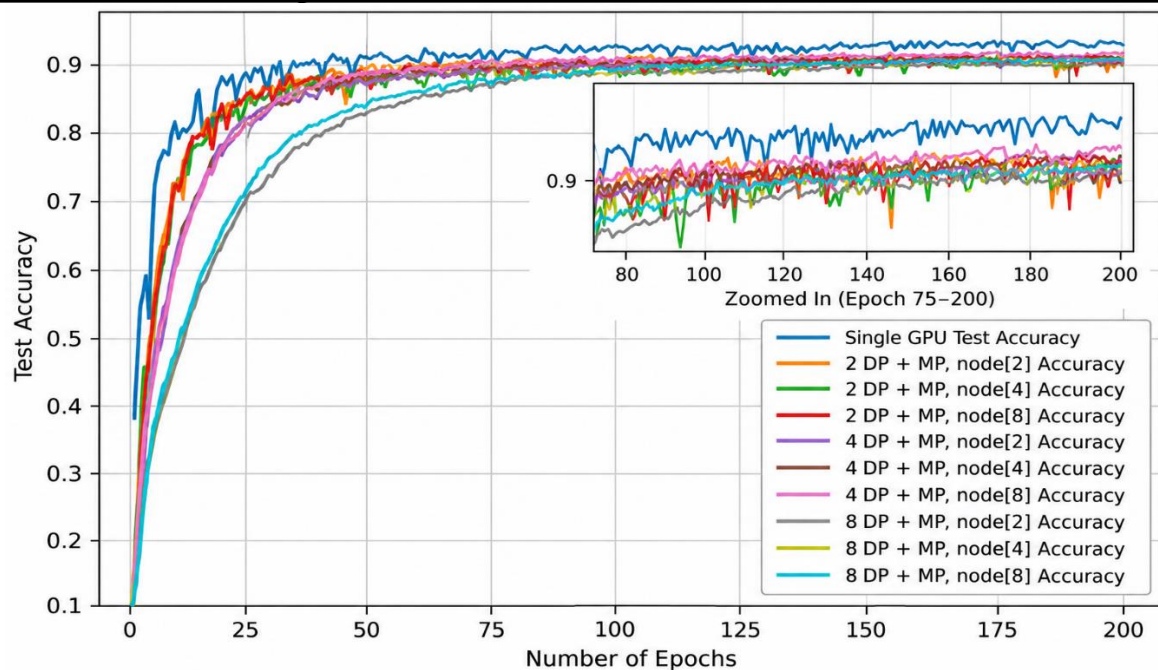


Figure 11: Test accuracy across training epochs under different GPU and distributed training configurations.

Overall, the results support the use of combined graph and temporal learning for the evaluated IoT-enabled ICS data. The proposed model outperformed its standalone components on macro-F1 while producing window-level predictions with a reported median inference time of 24 ms. The response module extended those predictions into context-informed action tiers and matched the intended tier in 95.1% of labeled scenarios. The chronological independent test period adds credibility to these findings by separating later observations from model development. The scope of the conclusions remains defined by the available results. Aggregate scores cannot show whether a rare or operationally severe class had a lower recall. The exact class counts, confusion matrix, observation-window duration, computing hardware, and tier definitions are needed to assess reproducibility and deployment implications. Evaluation on another ICS installation and on previously unseen attack patterns would also be needed to establish how well the reported performance generalizes beyond this dataset.

6- Future Work:

The present study evaluated the GAT-BiGRU framework on a chronologically separated dataset of 285,000 records covering normal

activity and six industrial cyberattack classes. Future work should examine whether its performance holds across different industrial facilities, device configurations, communication protocols, and operating conditions. Training on one environment and testing on another would provide a stronger measure of generalization than evaluation within a single dataset. Such testing should also include changes in device topology and previously unseen attack patterns [38]. Further analysis should focus on performance for each attack class. Although the framework achieved 97.3% macro-F1 on the independent test set, class-level precision, recall, confusion patterns, and detection delay would show where improvements are most needed. Rare or overlapping attack classes may require additional attention. Future experiments could also measure how performance changes with observation-window duration, graph construction rules, and the amount of data available immediately after an attack begins. The response module should be evaluated beyond agreement with labeled action tiers. Its reported 95.1% tier-assignment accuracy provides a basis for further study, but future work should examine whether tier errors lead to delayed escalation or unnecessary intervention.

Testing response recommendations in a controlled ICS testbed would allow their effects on process stability, service continuity, and operator workload to be assessed before operational deployment. Human review and site-specific safety procedures should remain part of that evaluation. Finally, real-time performance should be measured across the full monitoring workflow [39]. The reported median inference time of 24 ms per observation window covers prediction, while data collection, preprocessing, alert delivery, and response may introduce additional delays. Future work should report end-to-end latency, high-percentile inference times, computing requirements, and performance under sustained traffic. These evaluations would clarify the framework's suitability for deployment on resource-constrained industrial monitoring systems.

Conclusion:

This study presented a Graph Attention Network–Bidirectional Gated Recurrent Unit framework for real-time cybersecurity monitoring, multiclass intrusion detection, and adaptive response-tier assignment in IoT-enabled industrial control systems. The GAT learned relationships among communicating devices, while the BiGRU captured changes in network, authentication, and process activity over time. A risk-scoring module combined the predicted attack class, model confidence, and operational context to assign an intended response tier. The framework was evaluated on 285,000 time-stamped records representing normal activity and six cyberattack classes. Using a chronological 70% training, 15% validation, and 15% independent test split, it achieved 98.2% accuracy, 97.6% macro-precision, 97.1% macro-recall, and 97.3% macro-F1 on the test set. It also recorded a 1.8% false alarm rate and a median inference time of 24 ms per observation window. Macro-F1 exceeded the standalone GAT and BiGRU models by 3.4 and 2.7 percentage points, respectively. The response module selected the intended action tier in 95.1% of labeled scenarios. These findings show that combining device-interaction and temporal information improved multiclass detection in the evaluated dataset and linked intrusion predictions to context-informed response

decisions. The conclusions are limited to the tested environment and attack classes. Validation across other ICS installations, previously unseen attacks, and controlled response scenarios is needed to establish the framework's wider operational reliability.

REFERENCES:

- Villafranca, A., Thant, K. M., Tasic, I., & Cano, M. D. (2025). AI-enabled IoT intrusion detection: Unified conceptual framework and research roadmap. *Machine Learning and Knowledge Extraction*, 7(4), 115.
- Hemdan, M. N. (2026). DICS: Distributed Cybersecurity Architecture with Attention-Enhanced Temporal Correlation for Multi-Phase Attack Mitigation in SD-IoT Networks. *Journal of Intelligent Decision Making and Information Science*, 3(9s), 1432-1474.
- Orman, A. (2025). Cyberattack detection systems in industrial internet of things (IIoT) networks in big data environments. *Applied Sciences*, 15(6), 3121.
- Mahajan, U., Somasekar, J., & Neerugatti, V. (2026). RC-ADAPT: A Resilient and Continuous Adaptive Intrusion Detection Framework for CPS. *Engineering, Technology & Applied Science Research*, 16(5), 39828-39834.
- Ahmad, B., Muneer, M., Jillani, S. A., & Ahmed, S. (2026, January). Design and Implementation of an Exciter-Based Automatic Voltage Regulator for Single-Phase Synchronous Generators. In 2026 1st International Conference on Innovations in Information and Communication Technologies (IICT) (pp. 1-6). IEEE.
- Shafique, A., Wu, G., Saeed, M. S., & Javeed, D. (2026). X-SecureNet: A Security-Driven Intelligent Framework for Intrusion Detection in Smart Industrial Infrastructures. *Cluster Computing*, 29(3), 168.

- Alserhani, F. (2025). Intrusion detection and real-time adaptive security in medical IoT using a cyber-physical system design. *Sensors*, 25(15), 4720.
- Zahid, M., & Bharati, T. S. (2025). Enhancing cybersecurity in IoT systems: a hybrid deep learning approach for real-time attack detection. *Discover Internet of Things*, 5(1), 73.
- Rana, S., Bajwa, A., Tonoy, A. A. R., & Ahmed, I. (2025). Cybersecurity in Industrial Control Systems: A Systematic Literature Review on AI-Based threat Detection for SCADA and IOT Networks. Available at SSRN 5267824.
- Ahmad, B., Ali, S., Muneer, M., Fatima, A., & Abbas, N. (2025). Wind energy integration into the SAARC region: A comprehensive review of optimal wind sites for a sustainable super smart grid. *Wind Energy*, 3(2).
- Chandrashekhar, A. S., Sarala, D. V., Ambuja, K., Channarayappa, R. K., & Kumar, K. S. (2026). An intelligence framework for threat detection and response in cloud-IoT-assisted enterprise environments. *Bulletin of Electrical Engineering and Informatics*, 15(1), 546-556.
- Shtayat, M. B. M., Hasan, M. K., Sulaiman, R., Islam, S., & Khan, A. U. R. (2023). An explainable ensemble deep learning approach for intrusion detection in industrial internet of things. *IEEE Access*, 11, 115047-115061.
- Almalki, S. S. (2025). A Deep Learning-Based Framework for Real-Time Detection of Cybersecurity Threats in IoT Environments. *International Journal of Advanced Computer Science & Applications*, 16(3), 430.
- Ahmad, B., Majeed, M. K., Soomro, A. A., & Jillani, S. A. (2026, January). Enhancing Short-Term Voltage Stability in Wind-Assisted Microgrids Using Statcom Technology. In 2026 1st International Conference on Innovations in Information and Communication Technologies (IICT) (pp. 1-6). IEEE.
- Ahmadi-Assalemi, G. (2022). Anomalous behaviour detection for cyber defence in modern industrial control systems.
- Nedungadi, N., Sankaran, S., & Achuthan, K. (2025). Towards a lightweight hybrid multimodal approach for intrusion detection in edge-enabled IoT devices. *Cluster Computing*, 28(16), 1010.
- Prajapati, Y., Suthar, O. P., Chauhan, P. S., & Singh, S. K. (2026). Adaptive Hybrid Deep-Learning for Real Time Multi Class Threat-Recognition in Smart-City Cyber Physical Infrastructures. *Procedia Computer Science*, 282, 2612-2624.
- Ahmed, K. A., Ibrahim, A., Qudr, L. A. Z., Tawfeq, J. F., Al Falahi, Q., & Radhi, A. D. (2025, May). Developing Adaptive AI-Driven Cybersecurity Frameworks for Real-Time Detection of Polymorphic Malware in IoT Networks. In 2025 3rd International Conference on Business Analytics for Technology and Security (ICBATS) (pp. 1-8). IEEE.
- Mohy-Eddine, M., Guezzaz, A., Benkirane, S., Azrou, M., & Farhaoui, Y. (2023). An ensemble learning based intrusion detection model for industrial IoT security. *Big Data Mining and Analytics*, 6(3), 273-287.
- Srinivasan, M., & Senthilkumar, N. C. (2025). Intrusion detection and prevention system (IDPS) model for IIoT environments using hybridized framework. *IEEE Access*, 13, 26608-26621.
- Tanveer, M. U., Munir, K., Amjad, M., Zaidi, S. A. J., Bermak, A., & Rehman, A. U. (2024). Ensemble-guard IoT: A lightweight ensemble model for real-time attack detection on imbalanced dataset. *IEEE Access*, 12, 168938-168952.
- Alazab, M., Awajan, A., Alazzam, H., Wedyan, M., Alshaw, B., & Alturki, R. (2024). A novel IDS with a dynamic access control algorithm to detect and defend intrusion at IoT nodes. *Sensors*, 24(7), 2188.

- Ray, A. K., Sharma, R., & Pandey, S. K. Multiclass Cyber Attack Classification in Smart Home IoT Networks Using Ensemble Machine Learning with the MLEdgeIoT Dataset.
- Ahmad, B., Jillani, S. A., Rafique, M., & Soomro, A. A. (2026, January). Design and Monitoring of a Tree-Inspired Vertical Axis Wind Turbine for Efficient Urban Wind Utilization. In 2026 1st International Conference on Innovations in Information and Communication Technologies (IICT) (pp. 1-6). IEEE.
- Sekar, K., Balachandran, S., Sivamayivelan, K., & Parthasarathy, V. D. (2027). A Systematic Study on Adaptive and Explainable AI-Based Intrusion Detection Systems for Modern IoT Environments. In *Cybersecurity Risks in Organizational Digital Transformation and AI Infrastructure* (pp. 201-226). IGI Global Scientific Publishing.
- Fatima, M., Rehman, O., Rahman, I. M., Ajmal, A., & Park, S. J. (2024). Towards ensemble feature selection for lightweight intrusion detection in resource-constrained IoT devices. *Future Internet*, 16(10), 368.
- Eshmawi, A. A., Aldrees, A., & Alharthi, R. (2025). Smart framework for industrial IoT and cloud computing network intrusion detection using a ConvLSTM-based deep learning model. *Frontiers in Computer Science*, 7, 1622382.
- Gulzar, Q., & Mustafa, K. (2025). Enhancing network security in industrial IoT environments: a DeepCLG hybrid learning model for cyberattack detection. *International Journal of Machine Learning and Cybernetics*, 16(7), 4797-4815.
- Lakshmaiya, N. (2026). Hierarchical Temporal-Spatial Deep Learning Framework for Real-Time DDoS Detection in IoT-Enabled Multi-Energy Systems Security. *Sustainable Computing: Informatics and Systems*, 101425.
- Czeczot, G., Rojek, I., & Mikołajewski, D. (2024). Autonomous threat response at the edge processing level in the industrial internet of things. *Electronics*, 13(6), 1161.
- Wahab, S. A., Sultana, S., Tariq, N., Mujahid, M., Khan, J. A., & Mylonas, A. (2025). A multi-class intrusion detection system for DDoS attacks in IoT networks using deep learning and transformers. *Sensors*, 25(15), 4845.
- Khan, A., Li, Y., Shoukat, S., Adil, M., & Javeed, D. (2026, June). A Security-Aware and Explainable Framework for Reliable Intrusion Detection in Industrial IoT Environments. In 2026 2nd International Conference on Communication Networks and Smart Systems Engineering (ICCNSSE) (pp. 47-52). IEEE.
- Aljumaily, M., Abdullah, S., & Abd Alhasan, A. (2025). A Hybrid Intrusion Detection Framework for CyberPhysical Security in Smart Home/Smart City IoT Systems. *International Journal of Mechatronics, Robotics, and Artificial Intelligence*, 1(2), 63-73.
- Odeh, A., & Abu Taleb, A. (2023). Ensemble-based deep learning models for enhancing IoT intrusion detection. *Applied Sciences*, 13(21), 11985.
- Baladhandapani, R., Kasinayakanahally, B. N. R., Grandhi, S. H., Kalaikumaran, T., Abiodun, E. F., & Paneru, B. (2027). AI-Driven Cloud-Enabled Cybersecurity Framework for Intelligent IoT Intrusion Detection. In *Advancing Cybersecurity at the Intersection of Parallel Computing and Blockchain* (pp. 219-254). IGI Global Scientific Publishing.
- Ishtiaq, W., Zannat, A., Parvez, A. S., Hossain, M. A., Kanchan, M. H., & Tarek, M. M. (2025). CST-AFNet: A dual attention-based deep learning framework for intrusion detection in IoT networks. *Array*, 100501.
- Solanki, M., & Gupta, S. (2025). A novel intrusion detection framework using ensemble learning in MQTT IoT applications. *Annals of Mathematics and Artificial Intelligence*, 1-23.

Moustafa, N., Koroniotis, N., Keshk, M., Zomaya, A. Y., & Tari, Z. (2023). Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions. *IEEE Communications Surveys & Tutorials*, 25(3), 1775-1807.

Fraihat, S., Yaseen, Q., Sanjalawe, Y., Abu-Errub, A., Makhadmeh, S. N., & Al-Betar, M. A. (2026). Intrusion detection in industrial internet of things network using feature optimization and hybrid deep learning. *Discover Internet of Things*, 6(1), 34.

