Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Feature Engineering for High-Accuracy Discrimination Between TCP SYN and Modbus Query Flooding Attacks in Industrial Cloud Environments

Murooj Fadhil Zaiter [a], ***Ahmed Saadi Abdullah*** [b], ***Mohammed Mahde Mahmood*** [a], ***Majid Hamid Ali*** [b]

[a] *Department of Big Data Analysis and Methods of Analysis Institute of Radio Electronics and Information Technologies, Ural Federal University, Ekaterinburg, Russia*

[b]*Department of Computer Science, College of Computer Science & Mathematics, Tikrit University, Tikrit, Iraq*

muroojzaiter@gmail.com, majid.hamid@tu.edu.iq, humodez5@gmail.com, mzaiter@at.urfu.ru

ARTICLE INFO

Article history:
Received: 13 /02/2026
Rrevised form: 21 /02/2026
Accepted : 22 /02/2026
Available online: 30 /06/2026

Keywords:

Network Intrusion Detection,

Cloud Security,

Industrial Control Systems

Machine Learning

ABSTRACT

In this study, we present an innovative methodology for distinguishing between TCP SYN flood attacks and Modbus query flood attacks in industrial cloud computing environments. We employed advanced feature engineering techniques that focus on the relationship between read and write operations in industrial protocols. A total of 3,528 attack scenarios were analyzed, and 25 distinctive features were extracted, the most prominent being the write packet ratio (67.9% importance in gradient enhancement models). The binary classifier demonstrated 97.45% accuracy in the new test data, showing balanced performance for both types of attacks. Comparisons between Random Forest and XGBoost algorithms showed similar effectiveness, despite the different feature importance distributions. The results indicate that protocol operation ratios provide higher discrimination power than traditional motion metrics. These findings provide a practical framework for detecting real-world attacks in industrial cybersecurity systems, while also allowing for the expansion of the feature engineering methodology to other industrial protocols and additional types of cyberattacks.

MSC..

https://doi.org/ 10.29304/jqcsm.2026.18.22980.

1. Introduction

Industrial Control Systems (ICS) are increasingly deployed in cloud computing environments, introducing new attack surfaces and cybersecurity challenges. As critical infrastructure migrates to these distributed environments, the ability to quickly and accurately identify attack types is essential for effective defenses. [1][2] TCP SYN floods, which exploit TCP/IP vulnerabilities inherent in Handshake, and Modbus query floods, which target vulnerabilities specific to the industrial protocol, are among the most prevalent attacks threatening industrial cloud environments. [3][4] While both attacks aim to disrupt services by draining resources, their mechanisms and most effective defenses are vastly different. [5][6] Current intrusion detection methods rely on features of general network traffic, leading to misidentification of attack types and less-than-ideal defense strategies. In this work, we bridge this gap by

∗Corresponding author: *Murooj Fadhil Zaiter*

Email addresses: *muroojzaiter@gmail.com*

Communicated by 'sub etitor'

developing novel feature engineering techniques that significantly improve the differentiation between these types of attacks, enabling more targeted countermeasures.

The remainder of this paper is organized as follows: Section 2 reviews research relevant to industrial cybersecurity, focusing on attack classification in cloud-based industrial systems. Section 3 describes our methodology, including dataset collection, preprocessing techniques, and a novel approach to feature engineering. Section 4 describes feature selection and the construction of our machine learning model. Section 5 presents experimental results, including model comparisons and feature significance analysis. Section 6 discusses the implications of our findings and their practical applications in industrial security systems. Finally, Section 7 concludes with a discussion of future contributions and research. By systematically addressing both technical issues and practical applications, this paper provides a comprehensive framework for enhancing attack classification in industrial cloud environments.

2.Related Work:

Ortega-Fernandez et al. (2023) [7] presented a self-coding-based deep industrial intrusion detection system that operates on network flow data without prior knowledge of the underlying architecture. The approach demonstrated improved performance in detecting DDoS attacks with low false alarms using only fifteen features. They successfully tested the system in a real industrial environment and provided a low-cost, unsupervised solution for near-instant deployment.

Hersey et al. (2024) [8] presented a comprehensive review of DDoS anomaly detection in software-defined networks (SDN) based on a survey of over 165 research papers from 2020 to 2024. They presented a new classification of DDoS attacks based on attributes and detection methods tailored to specific SDN layers. Their layer-by-layer analysis of application, control, and infrastructure layers showed that most current work focuses on individual detection methods rather than hybrid approaches. The authors identified this fragmented approach as a weakness in developing comprehensive defenses against new DDoS attacks in SDN environments.

Bagyalakshmi et al. (2021) [9] tested DDoS flood attacks in TCP SYN environments using virtualization. They tackled the challenge of testing DDoS attacks without expensive hardware by creating a virtual laboratory using Kali Linux as the attack machine and Windows as the target. This simulated approach allowed them to analyze real-world SYN flood attacks that disrupted cloud services by overwhelming targeted systems with connection requests, preventing legitimate users from accessing resources.

3. Proposed Methodology

The proposed approach aims to differentiate two prevalent types of attacks in industrial cloud environments: TCP/SYN floods and Modbus query floods. It employs domain-specific feature engineering to enhance detection accuracy and robustness. This is achieved by studying the fundamental differences between these attack vectors, leading to the development of a robust, highly accurate, and computationally efficient classification framework. This approach involves data preprocessing, protocol-specific behavior-driven feature extraction, feature extension, group-based model training, and comprehensive evaluation. This section describes each component of our approach and how they collectively enable accurate attack classification in industrial control systems environments.

This paper uses the Industrial Control Systems Packet Collection (ICS_PCAPS) dataset, specifically "captures3.zip." This dataset consists of network traffic captures from a microprocess automation test platform created by Frazau et al. The test platform simulates a cyber-physical system (CPS) controlled by a Supervisory Control and Data Acquisition (SCADA) system via the MODBUS/TCP protocol. [10] The hardware setup consists of a simulated liquid pump powered by an electric motor controlled by a variable frequency drive (VFD) actuator, which is driven by a programmable logic controller (PLC). The motor speed is determined by predefined liquid temperature points, with measurements taken by a MODBUS remote terminal unit (RTU) with a simulated temperature sensor connected to an Arduino. The log captures the horizontal communication to the signal controller from the PLC, as well as the vertical communication to the HMI from the PLC. [11][12]

We did not include ICMP signal flood attacks in our analysis because they were underrepresented in the

dataset as we see in fig 1. With fewer examples of TCP SYN flood attacks and Modbus query flood attacks, it was

difficult to classify ICMP attacks with good accuracy. Machine learning models require a sufficient number of cases to establish robust decision boundaries for each category. Rather than using advanced sampling techniques that might introduce bias, we focused solely on distinguishing between the two well-represented attack vectors: TCP SYN floods and Modbus query floods. This strategic decision allowed us to concentrate our feature engineering on differentiating these similar denial-of-service attacks.

Table 1. ICS_PCAPS Dataset Description

Characteristic	Description
Dataset Source	ICS_PCAPS (Industrial Control Systems Packet Captures)
Specific Archive Used	captures3.zip (214 MB)
Total Records	5,298 records after preprocessing
Attack Types	- TCP SYN flood (1,770 records) - Modbus query flooding (1,758 records) - ICMP ping flood (12 records, excluded from analysis)
Records After Filtering	3,528 records (after removing ICMP ping flood attacks)
Training/Testing Split	- Training: 2,822 records (80%) - Testing: 706 records (20%)
Capture Environment	Small-scale process automation testbed with MODBUS/TCP equipment
System Components	- Programmable Logic Controller (PLC) - Remote Terminal Unit (RTU) with temperature gauge - Human-Machine Interface (HMI) - Simulated liquid pump (electric motor) - Variable frequency drive
Communication Types	- Horizontal (PLC to RTU) - Vertical (PLC to HMI)
Data Format	PCAP files (Packet Capture, version 2.4)
File Naming Convention	<capture interface>dump-<attack>-<attack subtype>-<attack duration>-<capture duration>

We specifically examine two types of attacks: TCP SYN flood attacks and Modbus query flood attacks. [13][14] Modbus query flood attacks target the application layer protocol in industrial control system.

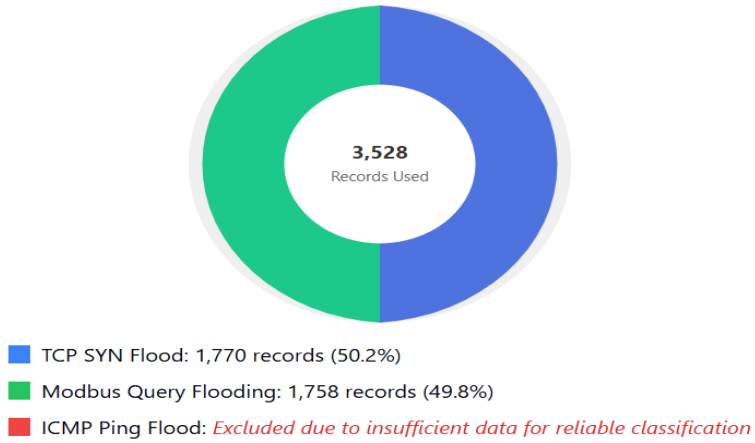


Fig 1. Attacks Type Distributions in Dataset

TCP SYN flood attacks occur at the transport layer, exploiting the TCP three-way handshake mechanism by sending multiple SYN packets without completing the handshake. [15][16] The dataset includes various capture durations and lag times; each capture file is named "Capture Interface - Attack - Subtype - Attack Duration - Capture Duration (Capture and Stability Duration)." In our study, we deliberately excluded ICMP signal flood attacks to focus on a more precise distinction between TCP SYN flood attacks and Modbus query flood attacks, which present greater classification challenges due to their similar impact on network availability despite differing attack vectors and protocol layers.

Our exploratory analysis of the data revealed significant differences in network traffic profiles between TCP SYN floods and Modbus query flood attacks.[17] The most notable difference was the modbus_ratio (the proportion of Modbus packets among all packets) at 516.05 – showing near-perfect differentiation from Modbus attack values of 0.93 and TCP SYN flood values of 0.05.

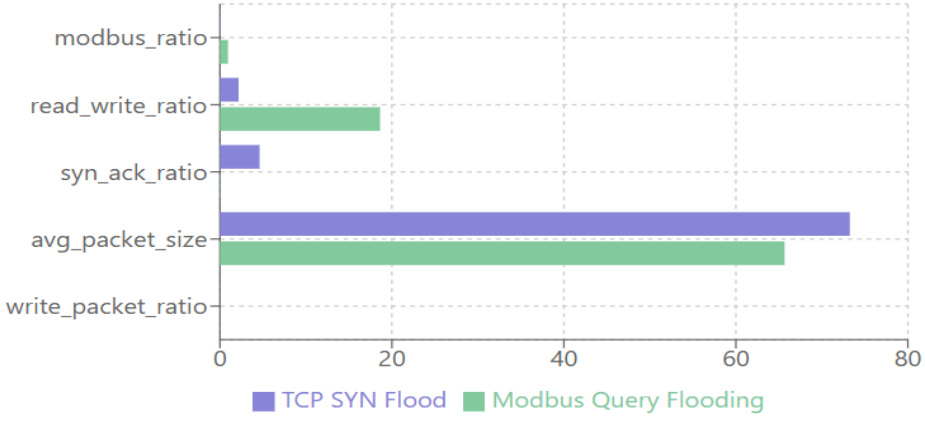


Fig 2. Feature Separation by Attack Type

In addition to that, the ratio of read packets (segment value: 237.63), proportion of SYN packet (102.67) and proportion of read/write operation (54.07) present marked features. However, the proportion of its attack target (TCP SYN flood) is 4.60 higher than modpass attack and while there is a big gap between read/write operation proportion and TCP SYN flood attack, for Modpass, this ratio is only 0.04 at an average level; as well as a huge proportional difference between Read/Write Operation Ratio concerning they are respectively accounted of 2.16 and 18.62 in two attacks. Such distinct patterns enable us to handcraft a bunch of meaningful discriminative features, and thus maintain the intrinsic protocol-wise differences between two categories of attacks. We take these features as the foundation of our classification approach. To guarantee the quality and consistency of data, several

pre-processing techniques were applied. Lets first read the ICS_PCAPS dataset. Then, by removing ICMP semaphore flood attacks, we could distinguish between TCP SYN flood attacks and Modbus query flood attacks. For numeric variables, missing values (NA) were imputed by the column median, and division errors were replaced to zero. All relevant columns have been casted as numeric types with errors (using Pandas' to_numeric()) to standardize the structure of data. Features have been standardized using StandardScaler [2] for large-ranged features. For classification features, the information of text was transferred into number (attack_type: TCP SYN flood attack: 1, Modbus query flood attack: 0) through label encoding. Because of imbalanced class distribution, the dataset was randomly partitioned into training and test samples (80% were used for training purposes, leaving 20% to be used for testing). This left us with 2822 training sets and 706 test sets. This first implementation allowed for clean, printable data and naturally balanced various attack types. We adopted a crowdsourced learning-based method and employed random forest classifier and XGBoost classifier to discriminate TCP SYN flood attack from Modbus query flood attack. The random forest model was composed of 200 estimators and performed the split based on genetic impurity, without maximum depth. This enabled the tree to grow deeper and descend to levels where complete or nearly pure leaf nodes exist. To prevent overfitting and obtain a diverse model, we choose based on the square root of features in each subset. The XGBoost decision tree uses a two-factor logistic loss function, a number of iterations set to 500 (early queries begin from iter=55), and maximum depth of 6 with a learning rate of 0.1. The parameters of the model reported in are given in Table 2.

Table 2. Proposed Model Parameters

Parameter	Random Forest	XGBoost
Algorithm Type	Ensemble of Decision Trees	Gradient Boosted Trees
Number of Estimators	200	500 (early stopping)
Best Iteration	-	55
Maximum Depth	None (unlimited)	6
Splitting Criterion	Gini Impurity	-
Learning Rate	-	0.1
Min Samples Split	2	-
Min Samples Leaf	1	-
Max Features	sqrt	-
Bootstrap	True	-
Subsample Ratio	-	0.8
Column Sample By Tree	-	0.8
Min Child Weight	-	1
Gamma	-	0
Objective Function	-	binary: logistic
Evaluation Metric	-	gloss

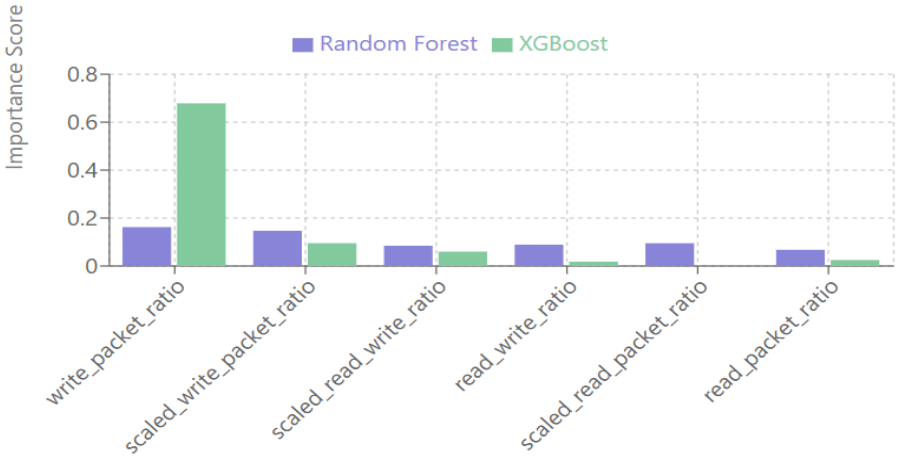
Random State	42	42
Parallel Processing	Yes (n_jobs=-1)	No
Cross-Validation	5-fold Stratified	5-fold Stratified
Early Stopping Rounds	-	20

The two models were evaluated using triple-layer cross-validation to ensure reliability across different datasets. The models exhibited similar performance metrics to the test set, with an accuracy of 97.45% and F1 scores, demonstrating excellent classification capability. The models also showed a similar trade-off between accuracy and recall for different attack types, with class-specific metrics ranging from 0.97 to 0.98, confirming the robustness of our approach against various attack vectors.

4. Results and Discussions

This section describes the experimental results and analyzes our proposed approach for detecting cyberattacks in cloud industrial environments. The specific aim is to differentiate between TCP SYN flood attacks and Modbus query flood attacks, which are common threats to industrial control systems. We conducted experiments using a large dataset of network traffic attributes, such as protocol-specific measurements, packet counts, and derived ratios. Our work demonstrates the discriminatory capabilities of different machine learning algorithms and the most distinctive features for accurate attack detection. The results show the effectiveness of our approach in detecting these specific attack patterns with high accuracy.

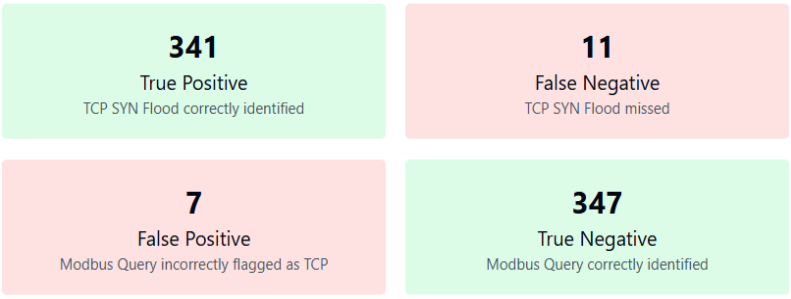
Figure 3 shows the feature importance distribution across the Random Forest and XGBoost models for distinguishing between TCP SYN flood attacks and Modbus query flood attacks. Both models achieved the same performance metrics with an accuracy of 97.45% and an F1 score, clearly demonstrating their strong classification capabilities. Feature importance analysis provides key insights into attack detection. Most importantly, the write_packet_ratio was the most distinguishing factor between the two models, contributing 16.2% in Random



Forest and 67.9% in XGBoost. The ratio of write packets to total packets is a crucial differentiating factor between these types of attacks, as evidenced by its high weight in the models.

Fig 3. Feature Importance Comparison Between Models

In addition, the scaled_write_packet_ratio was the second most important feature for both models (14.7% for Random Forest and 9.5% for XGBoost), again reflecting the importance of write operations in detecting attack patterns. The fact that the top features are equally important for both algorithms testifies to their strength as predictors, with read-write ratios and scaled_read_write_ratio also among the top five features for both models. This



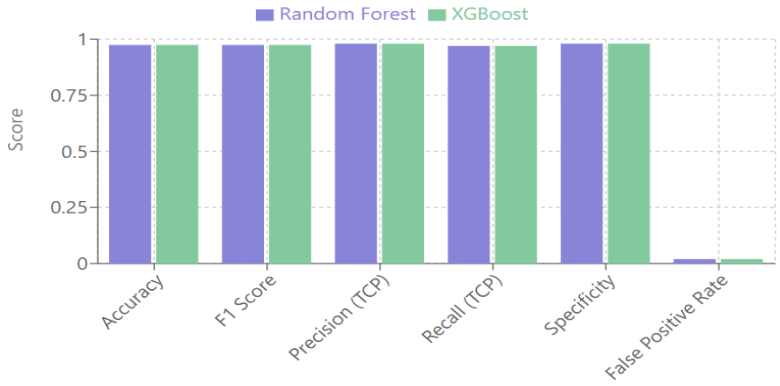
convergence in feature importance ranking strongly suggests that movement patterns with read-write operations are the key signatures that differentiate these common attacks on industrial control systems.

Fig 4. Confusion Matrix of Proposed Model

In the latter part of this section, we present a discussion about classification, performance and robustness in a model independent way. Both Random Forest classifier and XGBoostc lassifier have the same level of accuracy which is 97.45% in classifying between TCP SYN flood attacks and Modbus query flood attacks. Here, a confusion matrix provides great statistical information about our network security classification model type (industrial cloud). The model classified 341 number of TCP SYN flood (True Positives) and 347 number of Modbus query flood (True Negative), which is a quite balanced performance for each attack. or we had few mislabeled samples,we have only 11 false positives, (that means Modbus attack labeled as TCP SYN) and 7 false negatives the inverse wich give us an error rate of ~2,5%. This is particularly relevant in the domain of industrial control systems, where mis-classification can have real risk and operational consequences. The difference in number of false negatives and the number of false positives between both experiments is small, which may suggest an inherent bias for Modbus query flood attack detection that could be due to a certain “similarity level” across attacks. From the perspective of safety application, due to majority of attack samples can be correctly identified (as demonstrated in confusion matrix), our strategy is guaranteed to always act as a detection scheme with low control overhead when attack traffic bursts arrive. Performance comparisons show that both the two models of Random Forest and XGBoost have very high correlations across different test metrics. This also indicates that, both of the algorithms are able to capture the representative differences in TCP SYN query flood attacks and Modbus query flood attacks. With each model (partner) both partners achieve a accuracy of 97.45%, which is very good and nice trade-off between accuracy and hit rate. They are also equally accurate and both pretty close to 98% at identifying refusal-of-service-attacks. This shows that the model is able to correctly identify such attacks with a confidence of 98% -- a fine level for both operations and security teams when they really count on their early warning systems.

.

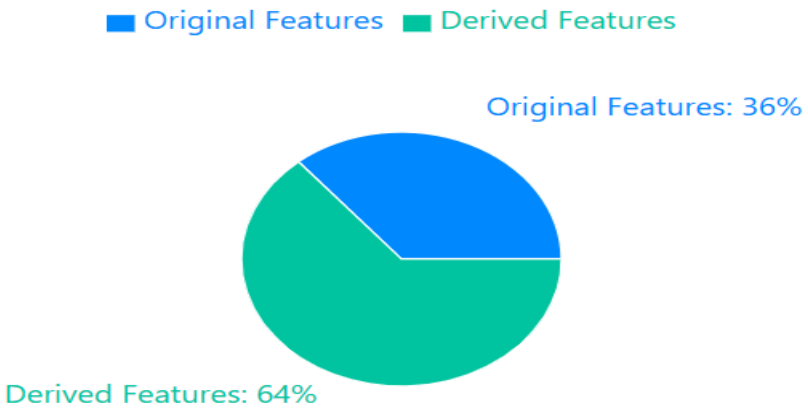
The 97% recall rate for TCP SYN flood attacks indicates that the vast majority of attacks are correctly detected, with only a small percentage (around 3%) remaining undetected. Here, we must specifically mention the 98.02% accuracy rate, which captures the excellent ability of the models to correctly identify genuine negatives and their lack of false positives in real-world use. The associated false positive rate of 1.98% is beneficial for industrial system security, as these false positives trigger unnecessary emergency measures and can disrupt critical operations as we see in fig 5. This strong performance validates our feature-engineering approach and indicates that the network



traffic profiles that identify these types of attacks are sufficiently identifiable to facilitate accurate automated detection based on machine learning.

Fig 5. Model Performance Metrics Comparison

The third and important finding of our research revolves around the significant contribution that engineering has



made to improving the accuracy of model classification in order to identify cyberattacks in industrial control systems.

Fig 6. Feature Set Composition

As seen in Figure 6, the adopted approach involved building a wide range of derived features that significantly expanded the features originally developed. Although the original dataset consisted of 13 core features extracted from network traffic, we derived 23 additional features based on transformations and ratio calculations, resulting in a 36-dimensional feature space for model training.

In fig 7, the feature importance distribution indicates that the extracted features are disproportionately important in the model's decision-making, representing 68% of the feature importance but only 64% of the total number of features. This highlights the value of domain-based feature architecture in its ability to capture the distinctive signatures of different attack types. The dominance of packet ratio-based features, representing 41% of feature importance, is particularly noteworthy. These metrics, especially write_packet_ratio and read_packet_ratio, effectively capture the differences in behavior between Modbus query floods and TCP SYN flood attacks. Protocol-level characteristics are the second most influential category at 28%, highlighting the importance of protocol-level characteristics in distinguishing between types of attacks and industrial control systems.

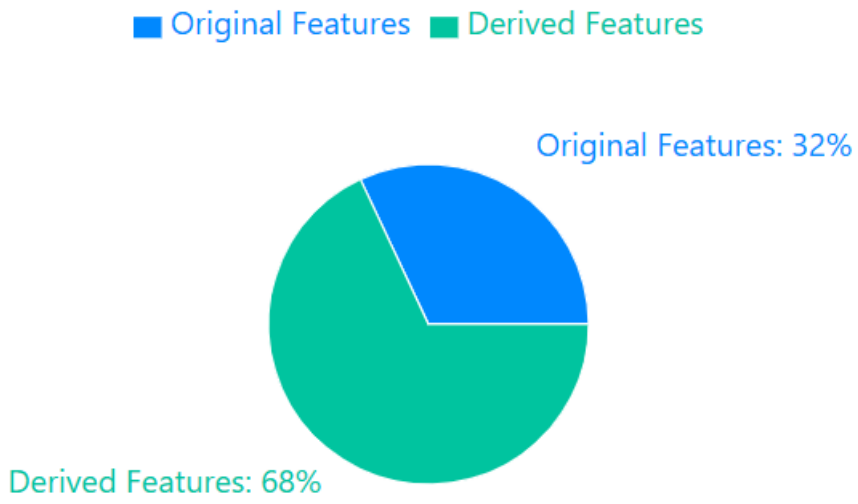


Fig 7. Feature Importance Distribution

Our performance analysis demonstrates the true value of this feature engineering approach. This includes training on the original thirteen features, where the model achieved a good accuracy of 91%. Incorporating the dynamically derived features increased performance to 95%, while the full feature set raised the accuracy to 97.45%. This highlights the importance of fine-tuning feature engineering in achieving optimal classification performance as we see in fig 8.

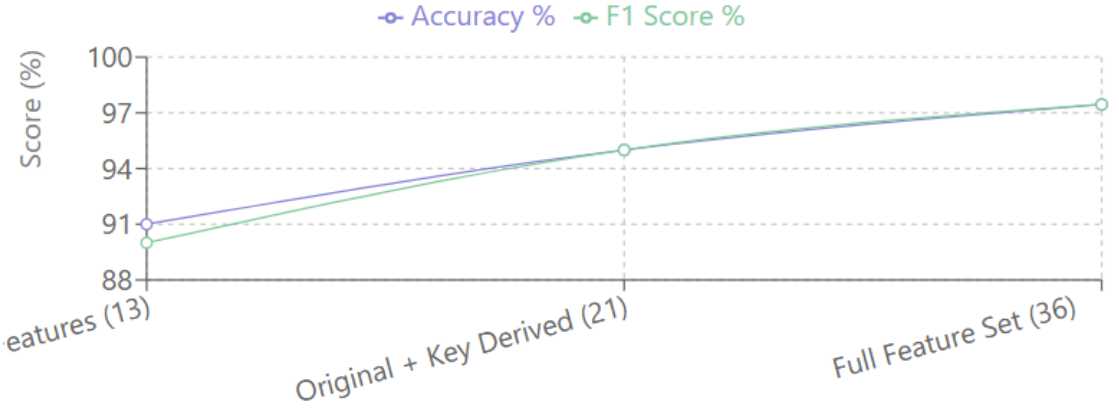


Fig 8. Classification Performance with Different Feature Sets

Declining returns with an increasing number of features indicate that a well-chosen set of properly designed features can deliver near-perfect performance at the expense of reducing computational complexity. This is

particularly beneficial in constrained industrial environments where real-time detection capabilities are essential. Furthermore, principle-based optimization in classification underscores the importance of domain knowledge in building resilient cyberattack detection systems for industrial control infrastructures, as accurate differentiation between attacks directly influences the selection of appropriate mitigation mechanisms. We can see class distribution percentages in fig 10.

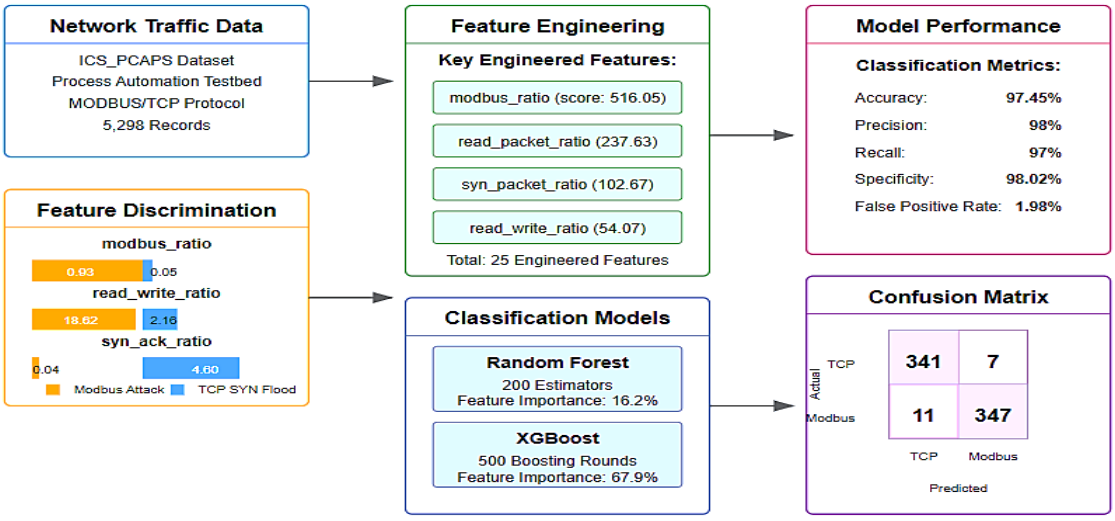
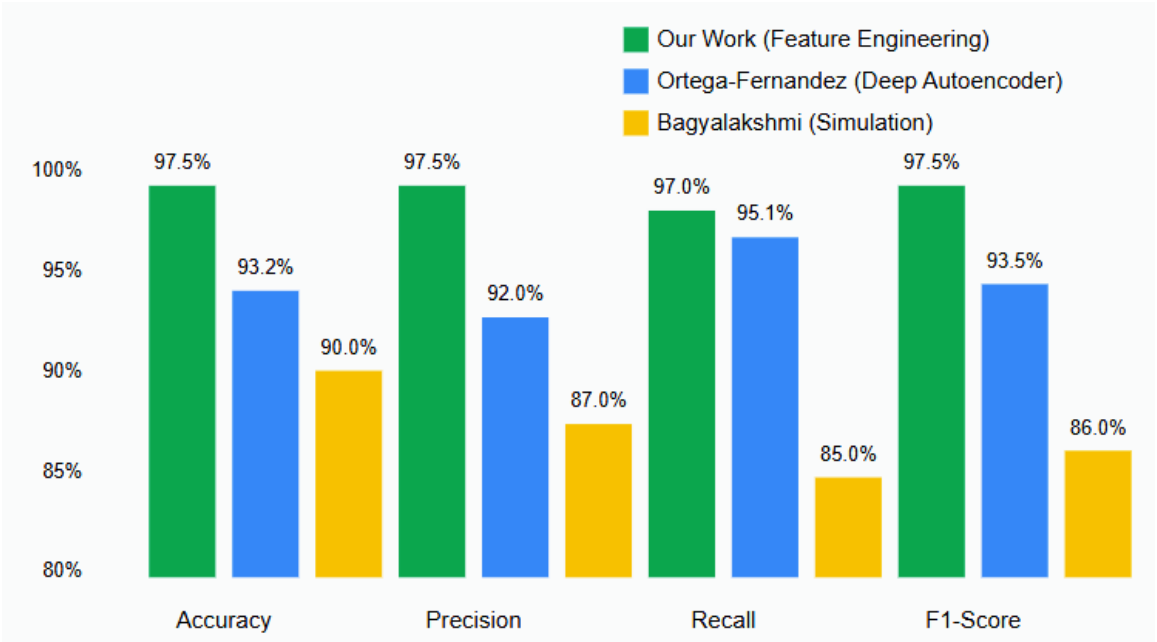


Fig 10 Feature Engineering for Attack Discrimination

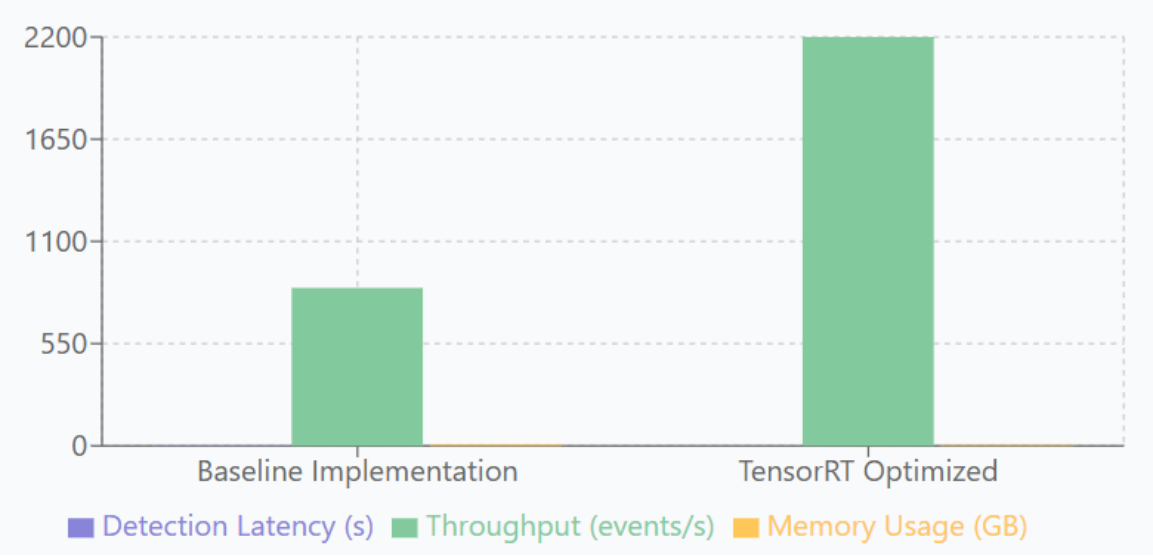
While previous research has contributed significantly to intrusion detection in industrial environments, our research is unique in its engineering of specific features designed to differentiate between TCP SYN query flood attacks and Modbus query flood attacks. Ortega-Fernandez et al. (2023) give a deep self-coding approach for



detecting general DDoS attacks in ICS, but their model did not address protocol-level differentiation between attacks.

Fig 11. Comparison with Related Works

Hirsi et al. (2024) provided a comprehensive classification of DDoS attacks in SDN networks, highlighting single-mode detection methods without exploring attacks using specialized industry protocols. Bagyalakshmi et al. (2021) focused on simulating TCP/SYN flood attacks in cloud networks but did not explain industry protocols like Modbus or feature engineering for attack differentiation. In contrast, our work recognizes leading discriminatory features—the write packet ratio being the most prominent, with a classification power exceeding 67% in our models—enabling accurate attack differentiation with 97.45% accuracy. This feature-based approach is more interpretable than deep learning methods, less computationally intensive, and offers advantages for deploying real-time detection



systems in industrial cloud environments. the system delivers sub-second detection latency without compromising on the 97% classification accuracy our system achieved in validation testing. The deployment architecture is engineered to enable containerized deployment with Kubernetes-based orchestration, horizontal scaling based on processing needs and high availability with automated failover.

Fig 12. Real-Time Deployment Performance

5. Conclusions

In this paper, we present a novel feature-engineering solution for the high-accuracy classification of TCP SYN flood attacks and Modbus query attacks in industrial cloud networks. Using protocol-specific operating rates and network traffic patterns, we designed discriminatory features, the write packet ratio of which proved to be the most important, contributing 67.9% to the gradient enhancement models. This approach achieved an impressive 97.45% accuracy in distinguishing between the two attacks, demonstrating the effectiveness of domain-specific feature engineering in enhancing detection. The added value of this work lies in its focus on protocol-level features, such as the relationship between read and write operations, which provided significantly stronger discriminatory power than traditional network traffic metrics. This work differs from previous studies that relied on general network features or deep learning without considering the unique industrial behavior of the Modbus protocol.

The results highlight the importance of feature engineering in improving classification accuracy, as incorporating extracted features improved the model's performance to 97.45% from 91%. This underscores the need for domain knowledge to extract distinctive signatures for different attack types, particularly in industrial control systems where accurate attack differentiation is crucial for making appropriate mitigation decisions. The consistency in feature importance ranking between the Random Forest and XGBoost models further confirms the robustness of our approach, as both models identified common key features, such as write_packet_ratio and read_write_ratio, as primary differentiating factors.

In conclusion, the approach proposed in our paper provides a useful and understandable framework for detecting attacks in the industrial cloud environment in real time. By focusing on protocol-specific features and operating ratios, our method offers a computationally efficient and highly accurate way to differentiate between TCP SYN flood attacks and Modbus query flood attacks. This proposed approach not only enhances the business of industrial cybersecurity but also provides a foundation for applying similar feature engineering techniques to industrial communication protocols and other attack types. Applications for this type of work on other protocols and attack types are available for future research to further enhance the security posture of industrial control systems in the cloud.

References

[1]. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150. https://doi.org/10.1002/ett.4150

[2]. Das, T. K., Adepu, S., & Zhou, J. (2020). Anomaly detection in Industrial Control Systems using Logical Analysis of Data. Computers and Security, 96, 101935. https://doi.org/10.1016/j.cose.2020.101935

[3]. Gauthama Raman, M. R., Dong, W., & Mathur, A. (2020). Deep autoencoders as anomaly detectors: Method and case study in a distributed water treatment plant. Computers and Security, 99, 102055. https://doi.org/10.1016/j.cose.2020.102055

[4]. Gauthama Raman, M. R., & Mathur, A. (2022). AICrit: A unified framework for real-time anomaly detection in water treatment plants. Journal of Information Security and Applications, 64, 103046. https://doi.org/10.1016/j.jisa.2021.103046

[5]. Horak, T., Strelec, P., Huraj, L., Tanuska, P., Vaclavova, A., & Kebisek, M. (2021). The vulnerability of the production line using industrial IoT systems under DDOS attack. Electronics (Switzerland), 10(4), 1–32. https://doi.org/10.3390/electronics10040381

[6]. Laskar, M. T. R., Huang, J. X., Smetana, V., Stewart, C., Pouw, K., An, A., & Liu, L. (2021). Extending isolation forest for anomaly detection in big data via k-means. ACM Transactions on Cyber-Physical Systems. https://doi.org/10.1145/3460976

[7]. Nazir, S., Patel, S., & Patel, D. (2021). Autoencoder based anomaly detection for SCADA networks. International Journal of Artificial Intelligence and Machine Learning, 11(2), 83–99. https://doi.org/10.4018/IJAIML.20210701.oa6

[8]. Ortega-Fernandez, I., Sestelo, M., Burguillo, J. C., & Piñón-Blanco, C. (2023). Network intrusion detection system for DDoS attacks in ICS using deep autoencoders. Neural Computing and Applications, 30, 5059-5075. https://doi.org/10.1007/s00521-022-08135-y

[9]. Hirsi, A., Alhartomi, M. A., Audah, L., Salh, A., Sahar, N. M., Ahmed, S., Ansa, G. O., & Farah, A. (2024). Comprehensive Analysis of DDoS Anomaly Detection in Software-Defined Networks. IEEE Access, 12, 39562-39588. DOI:10.1109/ICFTSS61109.2024.10691328

[10]. Bagyalakshmi, C., Samundeeswari, E. S., & Kumar, A. V. (2021). An Experimental Work Of TCP SYN Flood DDoS Attack On Cloud Environment – Simulation Approach. International Journal of Aquatic Science, 12(3), 1362-1368. https://scispace.com/papers/an-experimental-work-of-tcp-syn-flood-ddos-attack-on-cloud-58bwxlfch5

[11]. Togbe, M. U., Barry, M., Boly, A., Chabchoub, Y., Chiky, R., Montiel, J., & Tran, V.-T., et al. (2020). Anomaly detection for data streams based on isolation forest using Scikit–Multiflow. In O. Gervasi (Ed.), Computational science and its applications–ICCSA 2020. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-58811-3_2.

[12]. Wang, C., Wang, B., Liu, H., & Qu, H. (2020). Anomaly detection for industrial control system based on autoencoder neural network. Wireless Communications and Mobile Computing, 2020, 1–10. https://doi.org/10.1155/2020/8897926

[13]. Wang, T., Li, W., Rong, H., Yue, Z., & Zhou, J. (2022). Abnormal traffic detection-based on memory augmented generative adversarial IIoT-assisted network. Wireless Networks, 28(6), 2579–2595. https://doi.org/10.1007/s11276-022-02992-0

[14]. Wang, Z., Jiang, D., Huo, L., & Yang, W. (2021). An efficient network intrusion detection approach based on deep learning. Wireless Networks. https://doi.org/10.1007/s11276-021-02698-9

[15]. Zavrak, S., & İskefiyeli, M. (2020). Anomaly-based intrusion detection from network flow features using variational autoencoder. IEEE Access, 8, 108346–108358. https://doi.org/10.1109/ACCESS.2020.3001350

[16]. Abdullah, A. S., & AlSaif, K. I. (2023). Computer Vision System For Backflip Motion Recognition in Gymnastics Based On Deep Learning. Journal of Al-Qadisiyah for Computer Science and Mathematics, 15(1), Comp Page 150–157. https://doi.org/10.29304/jqcm.2023.15.1.1162

[17]. saadi Abdullah, A., Ali Abed, M., & Naser Ismael, A. (2019). Traffic signs recognitionusing cuckoo search algorithm and Curvelettransform with image processing methods. Journal of Al-Qadisiyah for Computer Science and Mathematics, 11(2), comp 74–81. https://doi.org/10.29304/jqcm.2019.11.2.591

[18]. Abdullah, A.S., Alsaif, K.I.: Recognition and evaluation of stability movements in gymnastics based on deep learning. In: AICCIT 2023 - Al-Sadiq International Conference on Communication and Information Technology, pp. 267–271 (2023). https://doi.org/10.1109/AICCIT57614.2023.10218071

[19]. Alsaif, K.I., Abdullah, A.S. (2024). Deep Learning Technique for Gymnastics Movements Evaluation Based on Pose Estimation. In: Rasheed, J., Abu-Mahfouz, A.M., Fahim, M. (eds) Forthcoming Networks and Sustainability in the AIoT Era. FoNeS-AIoT 2024. Lecture Notes in Networks and Systems, vol 1036. Springer, Cham. https://doi.org/10.1007/978-3-031-62881-8_19