Final Project Report
Cracking the Code: Investigating Advanced Intrusion Detection Frameworks for SCADA Security

Savali Sandip Deshmukh
sdeshmukh@ucdavis.edu

Shivani Suryawanshi
ssuryawanshi@ucdavis.edu

Tanvi Mehta
tanmehta@ucdavis.edu

December 12, 2024

1 Introduction

Supervisory Control and Data Acquisition (SCADA) systems are essential for managing critical infrastructure like manufacturing, energy distribution, and water treatment. By enabling real-time monitoring, SCADA supports informed decision-making for operational efficiency. However, increased internet connectivity makes SCADA systems vulnerable to cyber threats, including buffer overflows, cross-site scripting, and SQL injection, which can disrupt public safety and operations.These attacks can compromise the availability, confidentiality, and integrity of critical data, leading to disastrous consequences in essential services. Among the most commonly targeted protocols in SCADA environments is the Modbus protocol, widely used for industrial communication but lacking intrinsic security features. This vulnerability makes Modbus an attractive target for unauthorized access and malicious activities.To address these risks, Intrusion Detection Systems (IDS) have become essential in detecting and mitigating threats in SCADA networks.

The primary goal of this research project is to:

- Analyze and evaluate advanced intrusion detection techniques applied to SCADA systems using the Gas Pipeline Dataset. By comparing we aim to identify the most effective methods for enhancing SCADA system resilience and security.
- Examine the common types of cyber-attacks on SCADA systems, particularly those exploiting vulnerabilities within the Modbus protocol.

2 Literature Review

The paper "Intrusion Detection and Identification System Design and Performance Evaluation for Industrial SCADA Networks" [1] focuses on the design and evaluation of an intrusion detection and identification system (IDIS) particularly for SCADA networks used in industrial settings. The authors recognize the critical vulnerability of SCADA systems to cyber-attacks due to their integration of information technology with operational technology, often resulting in significant operational and safety risks. To address this, they propose an IDIS framework that utilizes machine learning algorithms to detect and classify different types of intrusions effectively. Their methodology involves detailed analysis and feature selection to enhance detection accuracy, while the performance of the proposed system is benchmarked using real-world SCADA network data. The results demonstrate high detection rates, indicating that the system can effectively distinguish between normal operations and various intrusion scenarios. Additionally, the study provides insights into the computational efficiency of the system, highlighting its potential application in real-time SCADA environments. This research contributes significantly to enhancing cybersecurity in critical industrial infrastructure.

The paper "A Stacked Deep Learning Approach to Cyber-Attacks Detection in Industrial Systems: Application to Power System and Gas Pipeline Systems" [2] presents a comprehensive study on enhancing the

detection of cyber-attacks in critical industrial systems through advanced machine learning techniques. The authors propose a novel, stacked deep learning framework that integrates multiple neural network models to improve the robustness and accuracy of intrusion detection. Focusing on power systems and gas pipeline infrastructures as case studies, the research highlights the growing need for effective cybersecurity measures in industrial environments vulnerable to complex cyber threats. The methodology involves leveraging deep learning's ability to capture non-linear relationships and subtle patterns in data to identify anomalies. The study's results show that the proposed approach outperforms traditional single-model systems, demonstrating superior performance in terms of both detection accuracy and false alarm reduction.

The paper "ICS-IDS: Application of Big Data Analysis in AI-Based Intrusion Detection Systems to Identify Cyberattacks in ICS Networks" [3] investigates the implementation of big data analytics in artificial intelligence-driven intrusion detection systems (IDS) for industrial control system (ICS) networks. The authors address the critical need for effective security mechanisms due to the increasing frequency and sophistication of cyber-attacks targeting ICS environments. Their proposed ICS-IDS framework leverages big data techniques to manage and analyze vast amounts of network data, ensuring timely and accurate threat detection. The study integrates AI models capable of distinguishing between normal and malicious activities, enhancing the system's capability to adapt to diverse attack patterns. Through extensive experimentation and performance evaluation, the results highlight the robustness and scalability of the framework in real-time intrusion detection.

3 Dataset

For this project, we plan to analyse the Gas Pipeline Dataset [4] for SCADA network security research created by Missouri University of Science and Technology (Missouri S & T). This dataset consists of 17 features and 2,74,628 instances across three class labels: binary, categorized, and specified. It includes 11 command payload features related to command injection attacks, 5 network features, and 1 response payload feature for response injection attacks.Some of the columns in the dataset are Timestamp, Source IP, Destination IP, Protocol, Packet Size, and Command Payload. The dataset covers 7 types of attacks which have been illustrated below in figure [1].

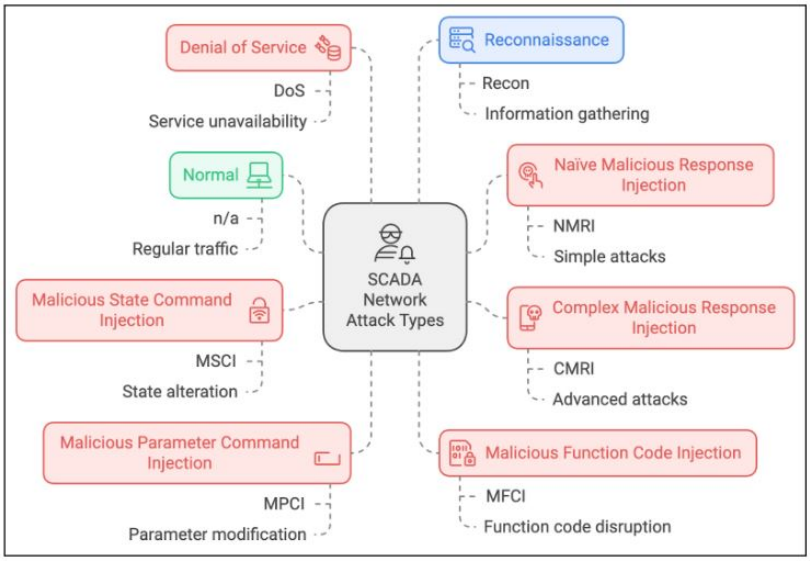

Figure 1: *Types of Attacks*

4 Methodology

The methodology for our Intrusion Detection System (IDS) on SCADA networks shown in figure [2] involves a structured workflow, as illustrated. We start with Data Collection and Preprocessing of the Gas Pipeline Dataset, which includes cleaning, handling missing values, and normalizing the data. After Data Splitting into training and testing sets, we proceed with Model Training using machine learning models such as Random Forest, Decision Trees, Support Vector Machines (SVM), and deep learning methods like Convolutional Neural Networks (CNNs), as used in previous research [1], [2], [3]. Next, we perform Hyperparameter Tuning to optimize model performance, followed by Model Testing on the test dataset. Finally, Performance Evaluation based on metrics like accuracy, precision, and recall to identify the most effective approach for intrusion detection on SCADA networks. This approach will enable a direct comparison of the different models' effectiveness.

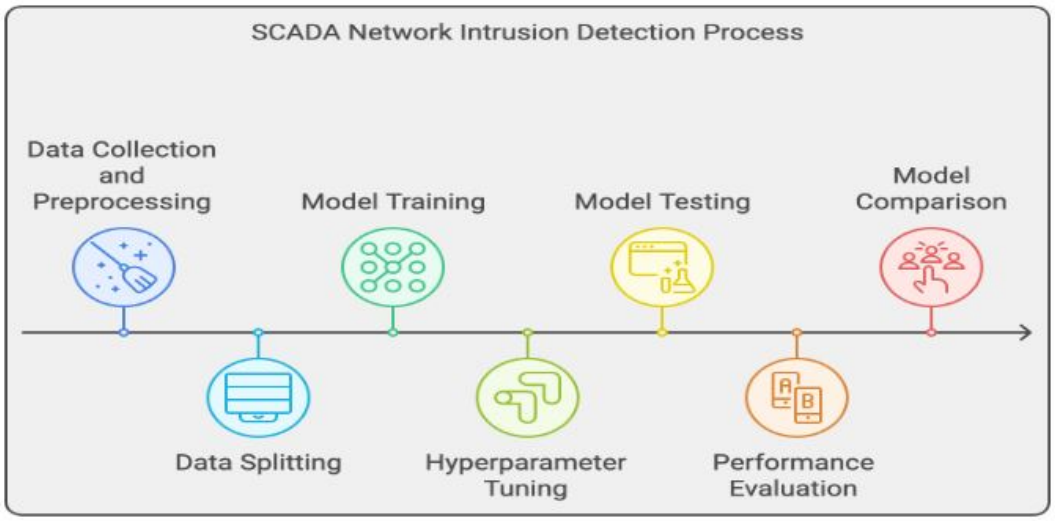

Figure 2: *Methodology*

5 Results

This section delves into the detailed results obtained from the analysis and implementation of methodologies described in three seminal papers addressing intrusion detection in SCADA systems. Each subsection highlights the reasoning behind model selection, preprocessing steps undertaken, the core methodology implemented, and the comparative results derived, including baseline performance metrics, confusion matrices, and accuracy scores. Github repository for results [5].

5.1 Results for Paper 1: Intrusion Detection Design [1]

5.1.1 Model Selection and Preprocessing

The Random Forest classifier was chosen for its proven ability to handle multi-class classification problems effectively, as highlighted in the paper. The preprocessing phase involved addressing missing values using KNN Imputation, ensuring that the dataset remained complete and consistent for training and testing. Key features such as "setpoint," "gain," and "pressure measurement" were preprocessed to handle missing entries, preserving the integrity of the dataset. The dataset was split into training and testing sets with a 67-33% ratio, adhering to standard practices for evaluating machine learning models.

5.1.2 Methodology Alignment

The methodology implemented was a baseline model reflecting the first stage of the proposed approach in the paper. The Random Forest model was trained to classify the dataset into its multiple attack categories,

including NMRI (Naïve Malicious Response Injection) and CMRI (Complex Malicious Response Injection). The training process included hyperparameter optimization to enhance the classifier's performance, ensuring it could generalize effectively on unseen data. This implementation in figure [3] serves as the foundational layer for intrusion detection and establishes a benchmark for future enhancements.

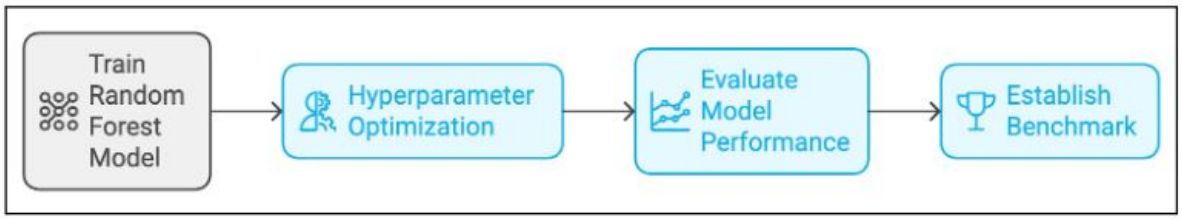

Figure 3: *Methodology Paper 1*

5.1.3 Results and Baseline Performance

The implemented Random Forest model achieved an accuracy of 94.95%, demonstrating robust performance in detecting various attack classes as in figure [4]. The confusion matrix revealed strong classification capabilities for most categories but highlighted challenges with certain overlapping classes, such as NMRI and CMRI. For instance, pressure measurement features contributed to confusion between these categories, as both exhibit similar patterns in specific ranges. Despite these challenges, the overall model performance remained high, with precision and recall scores consistently exceeding 0.90 for most classes.

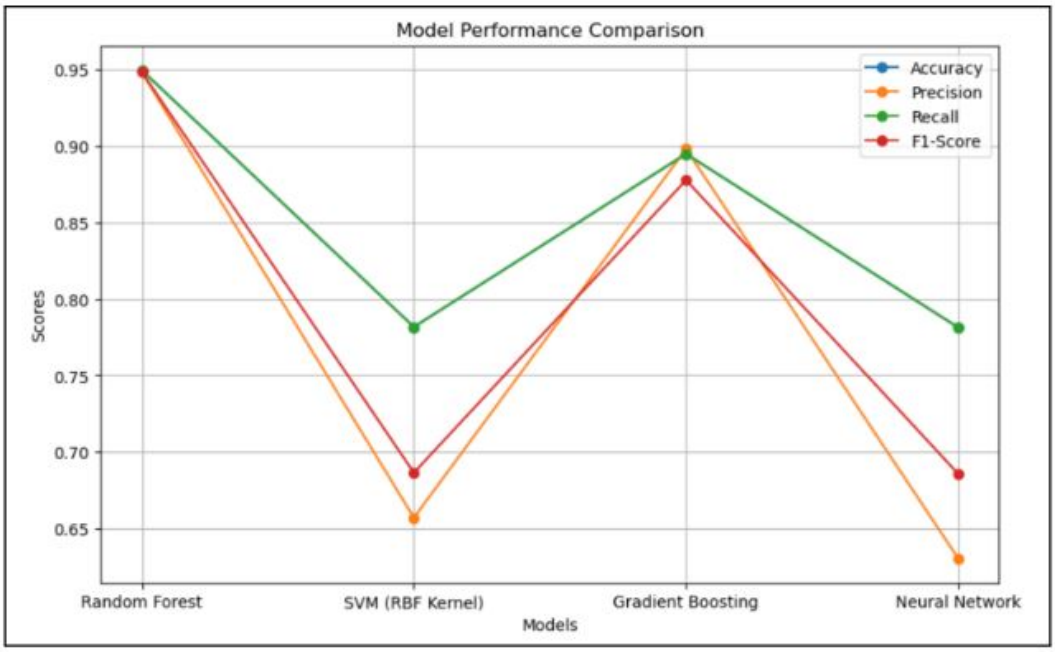

Figure 4: *Comparison Graph for Paper 1*

The baseline model's confusion matrix and classification report provided critical insights into its strengths and limitations. Key metrics such as weighted precision (0.95) and recall (0.95) underscore the model's reliability as a foundational intrusion detection system as represented in figure [5]. This implementation establishes a strong baseline for further experimentation, aligning closely with the goals of the referenced paper while leveraging the available dataset and preprocessing techniques effectively.

Model Comparison:

	Model	Accuracy	Precision	Recall	F1-Score
0	Random Forest	0.949563	0.948313	0.949563	0.948424
1	SVM (RBF Kernel)	0.781844	0.657251	0.781844	0.686619
2	Gradient Boosting	0.894900	0.898119	0.894900	0.877766
3	Neural Network	0.781480	0.630598	0.781480	0.685806

Figure 5: *Results Comparison for Paper 1*

5.2 Results for Paper 2: Stacked Neural Network [2]

5.2.1 Model Selection and Preprocessing

This meta-model was selected due to its ability to combine the strengths of multiple base models, enhancing overall prediction accuracy and robustness, particularly for imbalanced and complex datasets. The dataset underwent extensive pre-processing, including handling missing data, scaling, and encoding categorical variables. The key steps were:

1. Imputation: Missing numerical values were replaced with their column mean, and categorical values with their mode.
2. Feature Scaling: Applied StandardScaler to normalize numeric features.
3. Label Encoding: Encoded categorical variables using LabelEncoder.
4. Data Splitting: The data was divided into training and testing subsets using a 70:30 split, stratified on the target variable to ensure balanced classes.

5.2.2 Methodology Alignment

The paper implemented a Random Forest Classifier and a Stacked Neural Network as shown in figure [6], to analyze the dataset. For the neural network:

1. Three architectures ([64, 32, 16], [128, 64, 32], and [32, 16, 8]) were trained and stacked for enhanced predictions.
2. Optimized using the Adam optimizer with a learning rate of 10-3, training for 50 epochs. The models were evaluated on accuracy, confusion matrices, and feature importance (for Random Forest).

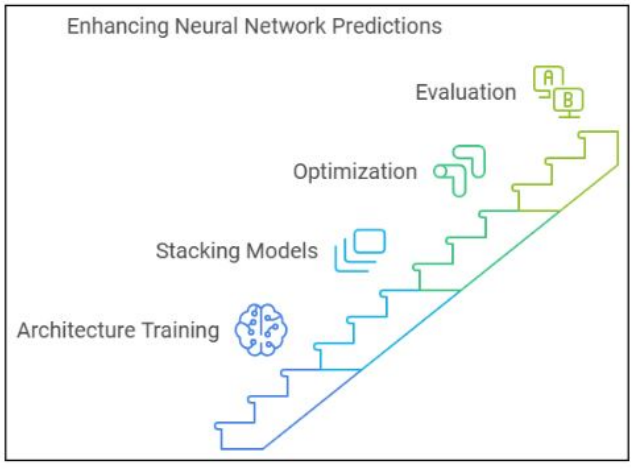

Figure 6: *Methodology Paper 2*

5.2.3 Performance Results

The stacked neural network model achieved a test accuracy of 86%, outperforming other configurations. Training and validation accuracy stabilized after 50 epochs.

The figure 7 presents a classification report summarizing the performance metrics of a meta-model applied to test data. The report details the precision, recall, F1-score, and support for each class. Class 0 achieved the highest recall (1.00) and an F1-score of 0.92, indicating excellent performance. However, class 1 showed poor results with zero precision, recall, and F1-score, reflecting challenges in detecting this class. The macro-average values (precision: 0.82, recall: 0.51, F1-score: 0.57) highlight imbalances across classes, while the weighted averages (accuracy: 0.86, weighted F1-score: 0.81) suggest strong overall performance despite inconsistencies in minority classes.

Meta-Model Classification Report (Test Data):

	precision	recall	f1-score	support
0	0.85	1.00	0.92	64374
1	0.00	0.00	0.00	2326
2	0.93	0.03	0.05	3911
3	0.91	0.31	0.46	2370
4	0.99	0.42	0.59	6124
5	0.92	1.00	0.96	1469
6	1.00	0.46	0.63	653
7	1.00	0.88	0.94	1162
accuracy			0.86	82389
macro avg	0.82	0.51	0.57	82389
weighted avg	0.84	0.86	0.81	82389

Figure 7: *Results Comparison for Paper 2*

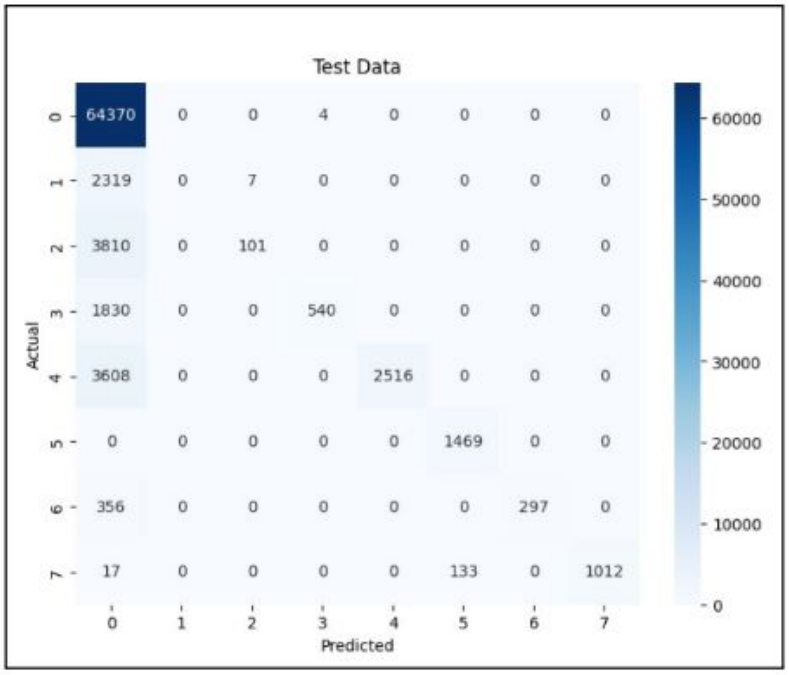

Figure 8: *Confusion Matrix*

The figure [8] is a confusion matrix visualizing the actual versus predicted classifications for the same test

data. The matrix shows dominant diagonal values, particularly for class 0, which indicates accurate predictions for this class. However, off-diagonal entries for classes like 1 and 2 reveal significant misclassification issues. The combination of these metrics and visualizations demonstrates the model's strengths in recognizing majority classes while identifying areas for improvement in handling underrepresented or complex classes.

5.3 Results for Paper 3: ICS-IDS [3]

5.3.1 Model Selection and Preprocessing

An AI-driven intrusion detection system was presented tailored for industrial control systems (ICS) networks. The proposed model addresses challenges of imbalanced multiclass datasets using a structured approach comprising data preparation and detection. Data preparation includes normalization, dimensionality reduction, and resampling through the All-KNN method to remove noisy and redundant samples, ensuring a balanced distribution between normal and attack data. The dataset was split into 70% training and 30% testing sets.

5.3.2 Methodology Alignment

The methodology implemented in the code aligns closely with the paper's proposed learning framework as in figure [9], focusing on two primary components: data preprocessing, feature engineering and model training and evaluation.

1. KNN: Configured with n_neighbors=5 to balance local sensitivity and generalization, achieving robust accuracy for both normal and attack traffic.
2. Random Forest: Utilized 100 estimators (n_estimators=100) with a random seed of 42 for consistent and high-performance classification, excelling in complex attack categories.
3. Naive Bayes: Demonstrated limitations in handling overlapping features due to its assumption of feature independence.
4. MLP (Multi-Layer Perceptron): Configured with one hidden layer of 100 neurons (hidden_layer_sizes=(100,)) and a maximum of 300 iterations (max_iter=300), delivering competitive results.
5. LSTM: Two layers of 50 units each (LSTM(50)) with softmax activation for multi-class classification, trained for 10 epochs with a batch size of 32. Effective in capturing patterns in time-series data.
6. GRU: Similar configuration to LSTM (GRU(50)) but optimized for faster convergence while maintaining comparable performance metrics.

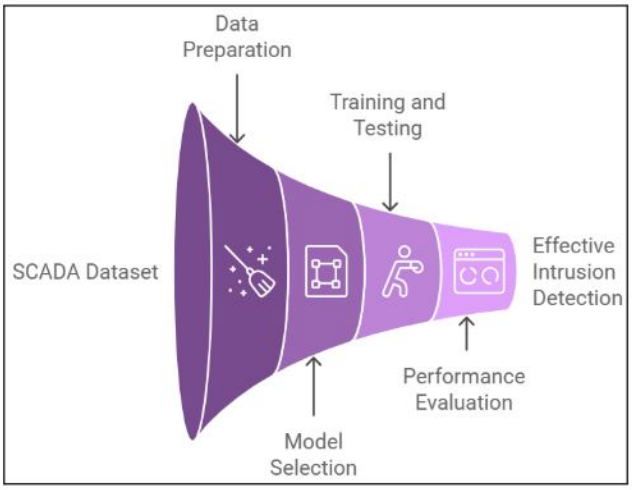

Figure 9: *Methodology Paper 3*

5.3.3 Performance Results

The Random Forest classifier demonstrated consistently strong performance across all label types. For binary labels, it achieved a high F1 score of 0.94 and an AUC-ROC of approximately 0.96, indicating its robustness in handling imbalanced datasets, as representated in figure [10]. Similarly, for categorized and specific labels, its macro-average F1 scores were around 0.91, making it the top performer among classical models. K-Nearest Neighbors (KNN) also performed well, especially for binary labels, with an F1 score of 0.93 and accuracy of 98% for categorized labels. However, it showed limitations in handling minority classes for specific labels. Naive Bayes struggled due to its strong assumptions, leading to poor results for categorized and specific labels, with overall accuracy falling below 10% for some cases. The Multi-Layer Perceptron (MLP) classifier offered competitive results for binary labels (F1 score of 0.82) but fell short on categorized and specific labels, highlighting its sensitivity to dataset complexity.

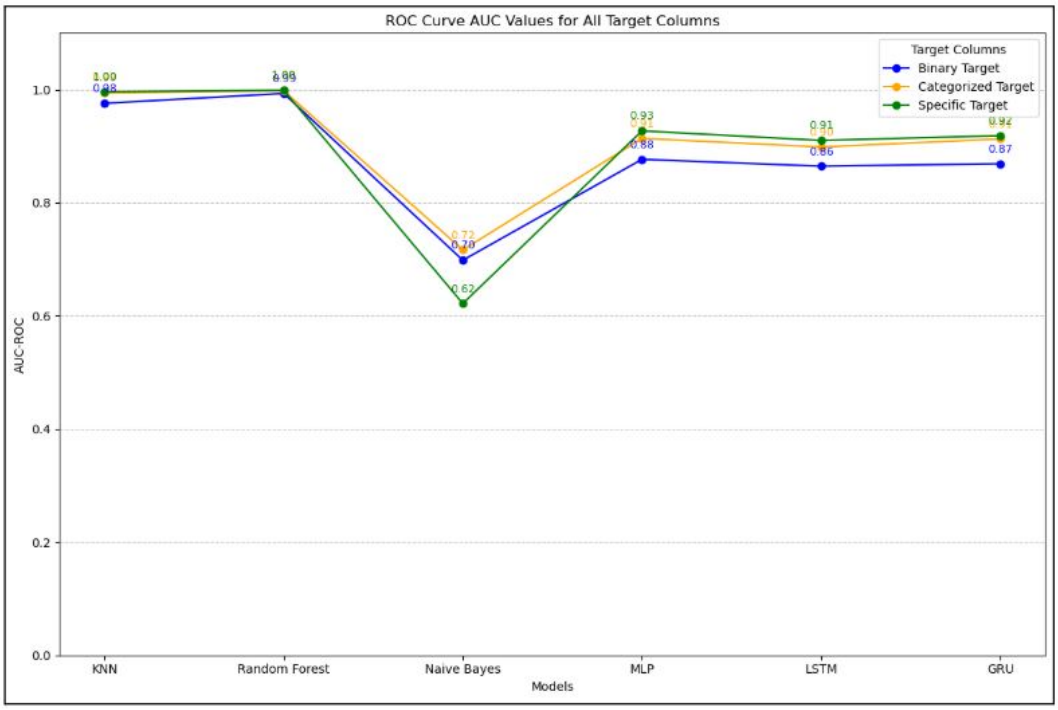

Figure 10: *ROC Curve AUC Values for Paper 3*

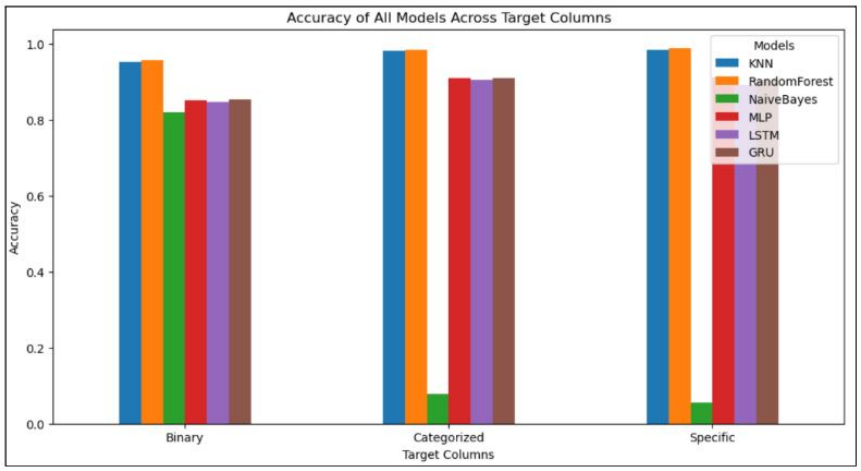

Figure 11: *Accuracy Graph for Paper 3*

LSTM and GRU models excelled in capturing sequential patterns, especially for categorized and specific labels. Both architectures achieved high accuracy (90%) and F1 scores (87–88%), demonstrating their suitability for complex, multivariate data. For binary labels, LSTM reached a test accuracy of 84.8%, and GRU slightly edged ahead with 85.4%, as demontrated in figure [11]. These results indicate that deep learning models can handle balanced and moderately imbalanced binary datasets effectively. Categorized labels saw GRU outperform LSTM marginally, achieving an accuracy of 91% compared to LSTM's 90.6%. The GRU's faster convergence also contributed to its slight edge in efficiency. Specific labels posed a greater challenge, but both LSTM and GRU delivered comparable results, with accuracy nearing 90.5%, and the AUC-ROC scores exceeded 0.91, reflecting their ability to generalize across classes.

5.4 Results and Discussion

The three methodologies explored in this project—Baseline Models, Stacked Neural Networks, and ICS-IDS—provided valuable insights into the efficacy of machine learning techniques for intrusion detection in industrial systems. Each approach utilized the same dataset and underwent preprocessing steps like imputation, encoding, and scaling, ensuring consistency across experiments. However, the evaluation strategies and architectures differed, which impacted performance metrics as shown in figure [12].

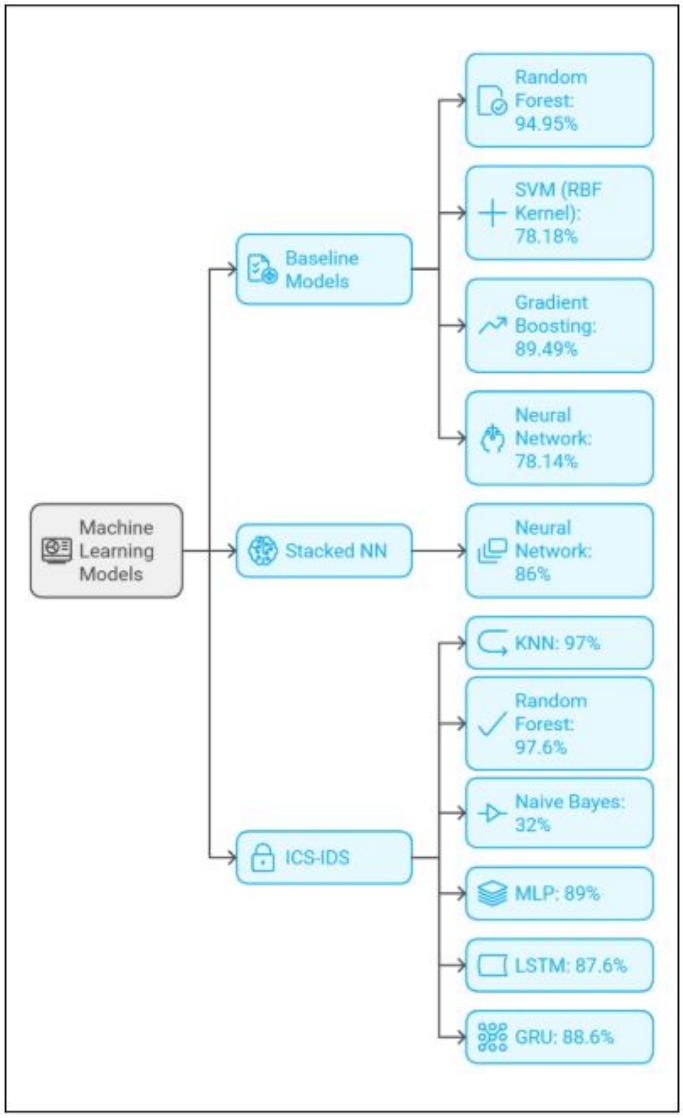

Figure 12: *Project Results*

1. **Baseline Models:** The baseline models established foundational performance benchmarks using standard machine learning techniques:
   (a) Random Forest: Achieved the highest accuracy in this group at 94.95%, showcasing its robustness in capturing complex patterns.
   (b) Gradient Boosting: Secured an accuracy of 89.49%, demonstrating strong predictive capabilities.
   (c) SVM with RBF Kernel: Performed modestly with 78.18%, likely limited by its inability to handle high-dimensional, complex feature spaces effectively.
   (d) Neural Network: Also scored 78.14%, indicating potential challenges in parameter optimization or architectural limitations in this implementation.
2. **Stacked Neural Networks:** The stacked neural network approach improved on the standalone neural network in the Baseline Models by leveraging multiple architectural configurations. The best-performing model achieved 86% accuracy, which, while not outperforming Random Forest in Baseline Models, highlighted the benefits of deep learning in capturing intricate patterns with suitable architectures.
3. **ICS-IDS Framework:** This method integrated a diverse set of machine learning models, achieving the highest accuracies across the board:
   (a) Random Forest: Outperformed all other models with 97.6%, reaffirming its dominance in handling imbalanced and complex datasets.
   (b) KNN: Delivered an impressive 97% accuracy, benefitting from its simplicity and effectiveness in small feature spaces.
   (c) MLP (Multi-Layer Perceptron): Achieved 89%, closely aligned with Gradient Boosting's performance in Baseline Models.
   (d) LSTM and GRU: Recorded 87.6% and 88.6%, respectively, emphasizing the utility of sequential models in temporal feature analysis.

Comparative Analysis:

1. Across all approaches, Random Forest consistently stood out as the top performer, particularly within the ICS-IDS framework.
2. The ICS-IDS methodology demonstrated clear superiority overall, likely due to its diverse ensemble approach and advanced preprocessing techniques.
3. Stacked Neural Networks improved upon the standalone neural network but did not surpass ensemble methods, suggesting that further architectural enhancements or hyperparameter tuning could yield better results.

Learnings and Insights:

1. Preprocessing consistency ensured comparability between methods, while variations in feature engineering (e.g., architectural stacking or ensemble selection) significantly influenced results.
2. Models like KNN and Random Forest excelled in simpler setups, while deep learning methods showed promise in scenarios with more extensive computational resources and fine-tuning.

Attack Class Observations:

Apart from the predominant class 0 (Normal traffic), the most common attack classes observed were NMRI (Naïve Malicious Response Injection) and Recon (Reconnaissance). Across the three methodologies, these classes were generally predicted with higher accuracy due to their distinct patterns compared to rarer or more complex attack types.

1. Baseline Models: Random Forest performed well in identifying NMRI and Recon, while models like SVM and Neural Network struggled with less frequent and complex attack classes such as CMRI and MFCI due to their sensitivity to imbalanced data.

2. Stacked Neural Networks: While improving over standalone neural networks, this methodology faced challenges in distinguishing between more nuanced attack classes like MSCI and MFCI. However, it consistently performed well on NMRI and Recon, owing to their simpler characteristics.

3. ICS-IDS Framework: This approach excelled across most attack categories, particularly NMRI and Recon, due to advanced preprocessing techniques that addressed class imbalance. However, some models, such as Naive Bayes, showed significant difficulty with complex classes like CMRI and MFCI, highlighting the importance of model selection for these categories.

Overall, NMRI and Recon were consistently well-predicted across methodologies, while CMRI and MFCI remained challenging due to their subtle patterns and lower representation in the dataset. This analysis reinforces the importance of matching model complexity to the dataset's characteristics and operational constraints.

6 Future Scope

Future research should focus on enhancing SCADA intrusion detection systems by integrating more advanced deep learning models, such as transformers or hybrid architectures. Emphasis should also be placed on developing explainable AI techniques to interpret model decisions, ensuring transparency and trust in critical infrastructure applications. Incorporating real-time anomaly detection capabilities with edge computing can further reduce latency, making SCADA systems more resilient. Further enhancements in preprocessing and feature engineering could improve detection of these rarer attack types. Additionally, expanding datasets to include more diverse and sophisticated attack scenarios will improve the robustness of intrusion detection models. Collaboration between academia, industry, and government can lead to standardized protocols for SCADA cybersecurity.

7 Problems Faced

Several key challenges were encountered throughout the project that significantly influenced the overall workflow and outcomes, spanning data preprocessing, computational constraints, and model performance optimization.

1. Data Preprocessing and Imbalance: Handling missing values in over 2,10,528 rows required extensive preprocessing using methods like KNN Imputation, StandardScaler, and LabelEncoder. Additionally, significant class imbalance and overlapping features, such as "pressure measurement" and "setpoint," led to challenges in distinguishing certain attack categories like NMRI and CMRI.

2. Computational and Time Constraints: The large dataset of 274,628 instances made training and testing time-intensive. Resource constraints limited the number of epochs for deep learning models like Neural Networks, LSTM, and GRU, impacting their optimization and overall performance. Iterative hyperparameter tuning further added to the time constraints.

3. Model Performance and Evaluation: Balancing model performance required careful hyperparameter tuning and evaluation across models while managing computational resources. This iterative process was essential to ensure fair comparisons and optimal results.

8 Conclusion

This research highlights the importance of robust intrusion detection systems for SCADA networks, emphasizing the critical role of Random Forest, LSTM, and GRU models in detecting cyber threats. Random Forest demonstrated excellent performance as a classical model, while deep learning models excelled in handling complex, sequential data. The findings underscore the vulnerabilities within SCADA environments, particularly in the Modbus protocol, and the necessity of tailored solutions to mitigate these risks. By analyzing various models and preprocessing techniques, this study establishes a strong baseline for future advancements, aiming to enhance SCADA system resilience against evolving cyber threats.

9 Appendix

Attack Type / Category / Class Name	Acronym	Brief Explanation of the names
Normal	n/a	Represents regular, non-malicious network traffic.
Naïve Malicious Response Injection	NMRI	Simple attacks that inject false responses into communication without complex behavior.
Complex Malicious Response Injection	CMRI	Advanced attacks that inject deceptive, sophisticated responses to disrupt communication.
Malicious State Command Injection	MSCI	Alters the state of devices by injecting unauthorized control commands.
Malicious Parameter Command Injection	MPCI	Injects malicious commands to modify parameters and operational settings of SCADA devices.
Malicious Function Code Injection	MFCI	Introduces unauthorized function codes to disrupt intended operations within SCADA protocols.
Denial of Service	DoS	Overloads the network or specific services to make them unavailable to legitimate users.
Reconnaissance	Recon	Probes the network to gather information on devices and configurations for potential future attacks.

Table 1: *Types of Attacks*

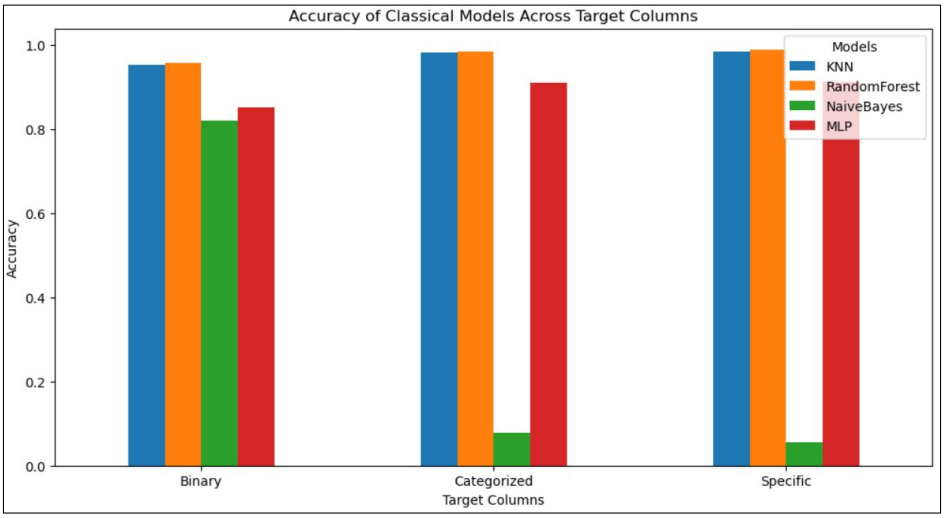

Figure 13: *ICS-IDS Framework Results - Classical Models Accuracy*

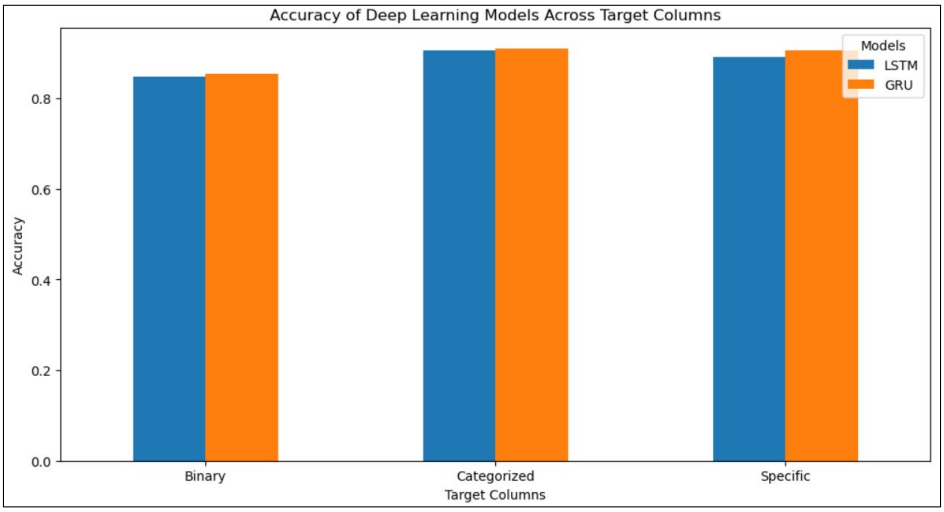

Figure 14: *ICS-IDS Framework Results - Deep Learning Models Accuracy*

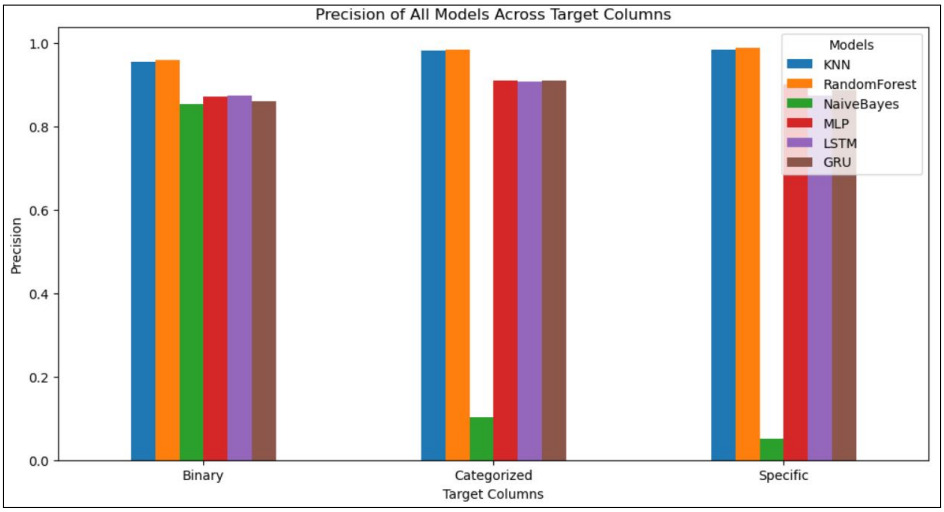

Figure 15: *ICS-IDS Framework Results - Precision Comparison*

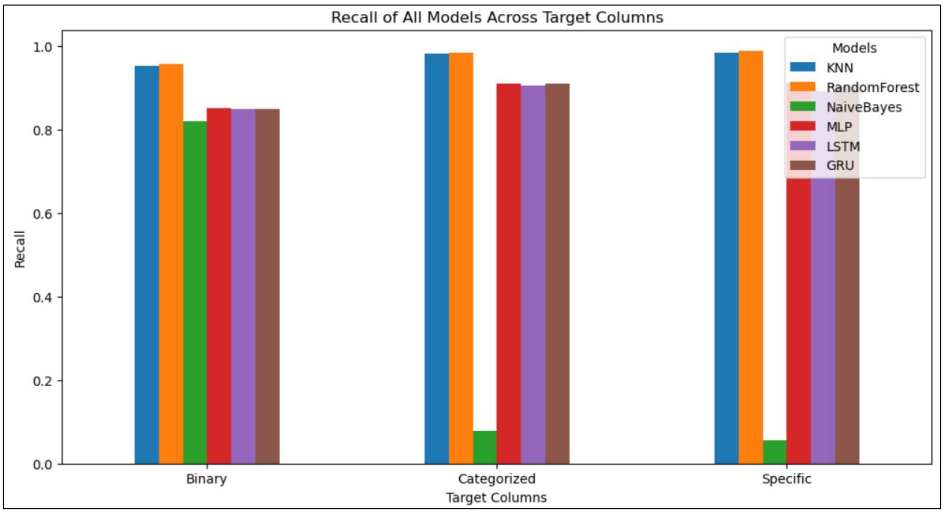

Figure 16: *ICS-IDS Framework Results - Recall Comparison*

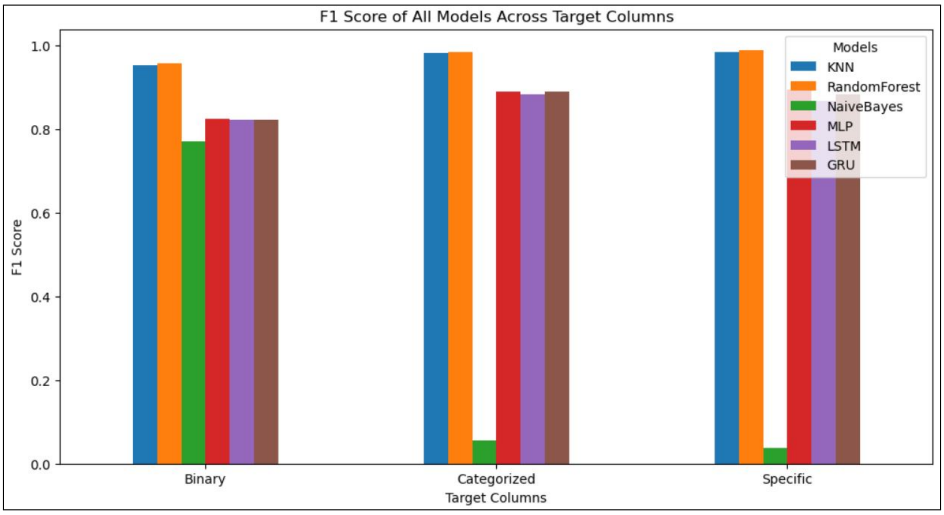

Figure 17: *ICS-IDS Framework Results - F1 Score Comparison*

References

[1] Ahsan Al Zaki Khan Gursel Serpen. "Intrusion Detection and identification System Design and Performance Evaluation for Industrial SCADA Networks". In: (2020). DOI: https://doi.org/10.48550/arXiv.2012.09707.

[2] Wu Wang Fouzi Harrou. "A stacked deep learning approach to cyber-attacks detection in industrial systems: application to power system and gas pipeline systems". In: *Springer Nature* (2021). DOI: https://doi.org/10.1007/s10586-021-03426-w.

[3] Bakht Sher Ali Inam Ullah. "ICS-IDS: application of big data analysis in AI-based intrusion detection systems to identify cyberattacks in ICS networks". In: *Springer Nature* (2023). DOI: https://doi.org/10.1007/s11227-023-05764-5.

[4] Thomas Morris Wei Gao. "Industrial Control System Traffic Data Sets for Intrusion Detection Research". In: *Springer Nature* (2014). DOI: https://doi.org/10.1007/978-3-662-45355-1_5.

[5] Tanvi Mehta, Shivani Suryawanshi, and Savali Deshmukh. *ECS235A-CIS-Cracking-the-Code-Advanced-Intrusion-Detection-Frameworks-for-SCADA-Security*. URL: https://github.com/tanvimehta11/ECS235A-CIS-Cracking-the-Code-Advanced-Intrusion-Detection-Frameworks-for-SCADA-Security.