ISSN NO: 9726-001X
Volume 13 Issue 02 2025

JOURNAL OF CURRENT SCIENCE

ANOMALY DETECION IN INDUSTRIAL CONTROL SYSTEM USING THE HAI SECURITY DATASET

Baddam Kavya[1] ,Puppala Rohith[2] ,Yedulapuram Abhiram[3] ,ana Mr. venkanna Mood [4]
[1,2,3] UG Scholar,Department of ECE,St.Martin's Engineering College,Secunderabad,Telangana,India, 500100
[4]Assistant Professor, Department of ECE,St.Martin's Engineering College,Secunderabad,Telangana,India, 500100
kavyareddy02697@gmail.com

Abstract:

Industrial Control Systems (ICS) play a crucial role in managing critical infrastructure, including power plants, water treatment facilities, and manufacturing units. With the growing interconnectivity of these systems, they have become increasingly vulnerable to cyber threats, leading to significant operational and financial risks. The Hyundai AutoEver AI (HAI) Security Dataset provides a comprehensive time-series dataset specifically designed for anomaly detection in ICS environments. Historically, ICS systems were built as isolated networks, relying on air-gapped security to prevent unauthorized access. However, with the evolution of Industry 4.0 and the integration of IoT devices, the systems have become more exposed to cyberattacks. Traditional security mechanisms, such as rule-based intrusion detection systems (IDS) and signature-based anomaly detection, have struggled to keep pace with the complexity of modern cyber threats. The systems are often ineffective against novel attacks and generate a high number of false positives, making real-time threat mitigation a challenging task. The primary problem lies in the detection of sophisticated attacks that manipulate sensor readings and actuator states to cause undetected disruptions in industrial processes. The HAI Security Dataset provides labeled time-series data collected from a realistic ICS testbed, making it highly suitable for machine learning (ML) and deep learning (DL)-based anomaly detection approaches. Traditional ICS security measures fail to leverage advanced data-driven techniques, leading to delayed response times and limited scalability in threat detection. The limitations of conventional security models highlight the urgent need for intelligent anomaly detection systems capable of learning dynamic patterns from real-world data. The significance of this study lies in the development of robust AI-driven models that can detect anomalies with high accuracy, reducing the risk of operational failures and security breaches in industrial environments. By utilizing the HAI dataset, this research aims to enhance ICS security through automated anomaly detection, thereby contributing to the resilience of critical infrastructure against emerging cyber threats.

Keywords:Industrial Control Systems (ICS), Cybersecurity, Anomaly Detection, Machine Learning (ML),Deep Learning (DL),Time-series Data, IoT Security, HAI Security Dataset, Sensor Manipulation

1. INTRODUCTION

Industrial Control Systems (ICS) are integral to the operation and management of critical infrastructure, including power grids, water treatment plants, oil refineries, and manufacturing industries. These systems rely on Supervisory Control and Data Acquisition (SCADA) and other automated control mechanisms to ensure seamless industrial operations. However, with the increasing digitalization and connectivity of ICS environments, particularly through the adoption of Industrial Internet of Things (IIoT) technologies, the risk of cyber threats and security breaches has significantly increased. Traditional ICS systems were initially designed as isolated environments with minimal cybersecurity considerations, relying on physical security measures and proprietary protocols for protection. However, the integration of networked systems and cloud-based solutions has introduced new vulnerabilities, making ICS a prime target for cyberattacks. Cyber threats targeting ICS can have severe consequences, including operational disruptions, economic losses, and even safety hazards in critical infrastructure. Traditional security mechanisms, such as firewall-based protection and rule-based anomaly detection, often fail to detect sophisticated attacks that manipulate sensor readings, actuator commands, or communication protocols. To address these challenges, machine learning (ML) and deep learning (DL)-based anomaly detection techniques have gained traction as effective solutions for identifying malicious activities in real-time. The Hyundai AutoEver AI (HAI) Security Dataset, specifically designed for ICS security research, provides real-world, labeled time-series data collected from a simulated industrial environment. This dataset enables the development and evaluation of intelligent models capable of detecting anomalies with high accuracy. The primary objective of this research is to design and implement an AI-driven anomaly detection framework leveraging the HAI Security Dataset. By analyzing sensor and actuator behavior, the model aims to distinguish between normal operations and cyber threats, ensuring early detection and mitigation of potential risks. This research holds significant importance in strengthening ICS cybersecurity by enhancing situational awareness, minimizing false positives, and enabling proactive threat responses. The proposed solution is expected to contribute to the resilience and reliability of industrial control

ISSN NO: 9726-001X
Volume 13 Issue 02 2025
JOURNAL OF CURRENT SCIENCE

environments, ultimately ensuring the safe and secure operation of critical infrastructure in an increasingly interconnected world.

2. LITERATURE SURVEY

Putchala et al. [1] proposed to apply a deep learning method using gated recurrent units (GRUs) to an intrusion detection system for IoT networks. The method showed a higher detection accuracy than traditional methods. They also proposed a lightweight and multi-layered design to enhance the security of IoT networks. Du et al. [2] proposed an unsupervised machine learning-based detection model based on LSTM-AE and GANs, which can learn complex patterns in time series data to detect anomalies more accurately. Goh et al. [3] introduced an unsupervised learning approach using RNNs to learn the changes in data patterns over time and use them to detect anomalies. In addition, Mokhtari et al. [4] used random forests to detect anomalous activity in industrial control systems. They showed that this method outperformed other classifier algorithms, which can significantly improve the detection of cyberattacks. Wolsing et al.[5] utilized random forests to effectively detect anomalous activity in industrial control systems. These techniques are proving to be highly effective in anomaly detection by learning complex data patterns and considering changes over time. Mahbod Tavallaee et al. [6] conducted a statistical analysis on the KDDCUP'99, finding that some issues with the dataset adversely affected the anomaly detection experiences. Gómez et al.[7] presented Electra8, an anomaly detection dataset for heterogeneous ICS scenarios. They selected the railway industry, and the Electra dataset was conducted using network traffic generated from normal and attack situations at a traction substation. Faramondi et al. [8] was used to generate an intrusion detection dataset for ICSs. They emulated water flowing between 8 tanks as Hardware in a Loop as a simulation tool to simulate the control system and networking infrastructure. Ferrag et al. [9] introduced the Edge-IIoTSet, a proposed dataset for Cyber Security in Internet of Things (IoT) and IIoT devices. The dataset encompasses a wide range of IoT devices and incorporates an extensive list of features derived from diverse sources such as alerts, system resources, logs, and network traffic.

Alsaedi et al. [10], comprises Telemetry data of IoT/IIoT devices collected in a controlled environment during both normal operations and in the presence of different cyber-attacks. In addition, the dataset also includes operating systems logs (such as disk or memory usage and process information) and network traffic of an IoT network, acquired from a realistic representation of a medium-scale network at the Cyber Range and IoT Labs. Ozay et al. [11] proposed an attack detection model employing state vector estimation (SVE) to detect false data injection at the physical layer of a smart grid. They showed that the model performs accurately on various IEEE test systems in detection of abnormal behaviors; however, it cannot detect the stealthy malicious activities properly. Pan et al. [12] introduced an IDS strategy leveraging features of signature-based and specification-based detection methods which protects an electrical power transmission line from attacks. Choi et al. [13] presented an IDS based on voltage measurement data to detect in-vehicle controller area network intrusions using inimitable characteristics of electrical signals.

3. PROPOSED METHODOLOGY

3.1 Overview

Industrial Control Systems (ICS) play a critical role in managing infrastructure in industries such as power plants, water treatment facilities, and manufacturing. These systems rely on Supervisory Control and Data Acquisition (SCADA) to monitor and control industrial processes. However, with the increasing integration of IT and OT (Operational Technology), ICS environments have become highly vulnerable to cyber threats, making anomaly detection essential to ensure security and operational continuity. Traditional security methods, such as rule-based intrusion detection systems (IDS) and firewall-based protection, are often inadequate in detecting sophisticated attacks. These methods rely on predefined rules or known attack signatures, making them ineffective against zero-day threats and evolving cyber-attacks.

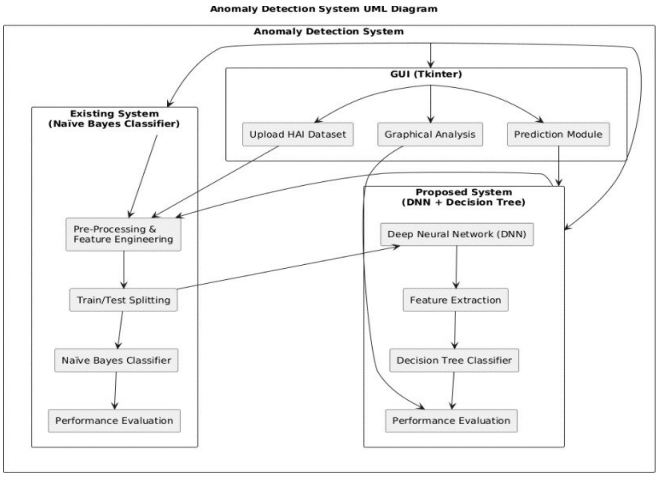

Fig 3.1 Proposed DNN with DTC system architecture

3.1.1 Objectives

1. **Develop an AI-Based Anomaly Detection System**

- Identify and classify abnormal behaviors in industrial control systems using **machine learning** and **deep learning** techniques.

2. **Enhance Security in Industrial Systems**

- Detect cyber threats, system faults, or any anomalies that could compromise operational safety using the **HAI Security Dataset**.

3. **Feature Engineering for Improved Prediction**

- Extract meaningful features from time-series data to improve classification accuracy.

4. **Compare Machine Learning and Deep Learning Approaches**

- Evaluate **Naïve Bayes Classifier** vs. **DNN + Decision Tree** to determine the best-performing anomaly detection model.

5. **Provide a GUI-Based User Interface**

- Develop a **Tkinter-based** application for easy dataset loading, model training, anomaly detection, and result visualization.

Key Components :

1. Data Preprocessing

- Handle missing values, normalize numerical features, and extract time-based attributes.

2. Feature Engineering

- Extract meaningful insights from raw sensor data (e.g., timestamp-based features).

3. Model Training & Comparison

- **Naïve Bayes Classifier (Existing System)**
    - Uses a probabilistic approach for anomaly detection.
- **DNN + Decision Tree (Proposed System)**
    - **Deep Neural Network (DNN)** extracts high-level features.
    - **Decision Tree Classifier** uses these features for final classification.

4. Performance Evaluation

- Metrics: **Accuracy, Precision, Recall, F1-score, and Confusion Matrix**
- Compare traditional ML vs. deep learning approaches.

5. GUI Implementation (Tkinter)

- **Dataset Upload**: Load the HAI dataset for training and testing.
- **Model Training**: Train models using selected algorithms.
- **Prediction Module**: Perform real-time anomaly detection on new data.
- **Graph Visualization**: Display performance comparison of different models.

3.2 Proposed workflow

The proposed workflow for anomaly detection in industrial control systems using the HAI Security Dataset begins with data preprocessing, where raw sensor data undergoes cleaning, normalization, and feature extraction to ensure high-quality inputs for the model. Next, in the feature engineering phase, relevant time-series features are extracted to enhance anomaly detection accuracy. The dataset is then split into training and testing sets, followed by the model training phase, where a Deep Neural Network (DNN) is combined with a Decision Tree Classifier to detect anomalies. This proposed approach is compared against the existing Naïve Bayes Classifier to assess improvements in detection accuracy. Once trained, the model undergoes performance evaluation using metrics such as accuracy, precision, recall, and F1-score to determine its effectiveness. The final stage involves GUI-based implementation using Tkinter, where users can upload datasets, train models, visualize performance graphs, and detect anomalies in real-time. This structured workflow ensures a robust and scalable AI-powered anomaly detection system for industrial control security.

3.3 Model Building & Training

The model development for **Anomaly Detection in Industrial Control Systems using DNN and Decision Tree Classifier (DTC)** consists of the following key stages:

1. Data Preprocessing & Feature Engineering

- **Dataset Used:** HAI Security Dataset
- **Data Cleaning:** Handle missing values, remove outliers, and normalize sensor data.
- **Feature Scaling:** Normalize numerical values to ensure uniformity for deep learning processing.
- **Feature Selection:** Select relevant features based on correlation analysis to reduce redundancy.
- **Data Splitting:** Divide data into **training (70%)**, **validation (15%)**, and **test (15%)** sets.

2. Deep Neural Network (DNN) Model for Feature Extraction

- **Input Layer:** Takes sensor values as input.
- **Hidden Layers:** Multiple dense layers with ReLU activation to capture non-linear patterns.
- **Dropout Layers:** Added to prevent overfitting.
- **Output Layer:** Produces a **feature vector** representation.

Feature Vector Extraction: The output of the last dense layer is treated as the extracted feature set for classification.

3. Decision Tree Classifier (DTC) for Classification

- The feature vector from the DNN model is passed to the Decision Tree Classifier.

- The Decision Tree learns patterns and splits data into:
    - Anomaly (Attack detected)
    - Normal (No threat detected)

4. Model Training

- **DNN Training:**
    - Optimizer: Adam
    - Loss Function: Categorical Cross-Entropy
    - Batch Size: 32 or 64
    - Epochs: 50+ (until convergence)
- **Decision Tree Training:**
    - Training on the feature vector extracted from the DNN.
    - Splitting criteria: Gini impurity or entropy.
    - Pruning techniques to prevent overfitting.

5. Model Evaluation & Testing

- DNN Feature Extraction Performance:
    - Loss vs. Accuracy Curve
    - Feature importance visualization
- **Decision Tree Classification Performance:**
    - Metrics: Accuracy, Precision, Recall, F1-Score, ROC Curve
    - Confusion Matrix for anomaly detection insights.

6. Deployment & Real-Time Monitoring

- The trained model is deployed in an industrial setting for real-time anomaly detection.

Continuous model updates & retraining using new data for improved performance.

3.3.1 Proposed DNN with DTC Model

In this hybrid model, DNN is used to learn hierarchical representations from the raw input data. The network comprises several dense (fully connected) layers with ReLU activations and concludes with a softmax layer for initial class prediction. However, to leverage the DNN's powerful feature extraction, features are taken from an intermediate layer and fed into a Decision Tree classifier. This ensemble method aggregates the predictions of multiple decision trees, which often results in improved robustness and accuracy compared to using the DNN alone.

DNN Component (Feature Extractor): The DNN is a fully connected feed-forward network that begins with an input layer accepting the preprocessed feature vector (with a dimension equal to the number of input features). It then processes the data through several hidden layers:

- The first hidden layer consists of **128 neurons** with a ReLU activation function, which begins the process of learning non-linear relationships.
- The second hidden layer reduces the dimensionality to **64 neurons**, further abstracting the data while preserving essential patterns.
- The third hidden layer contains **32 neurons**, continuing to distill the information into a more compact form.
- The fourth hidden layer, with **16 neurons**, serves as the final stage of feature abstraction.
- A final dense layer with **8 neurons** and a softmax activation is originally intended for classification purposes when the DNN is used as a standalone model.

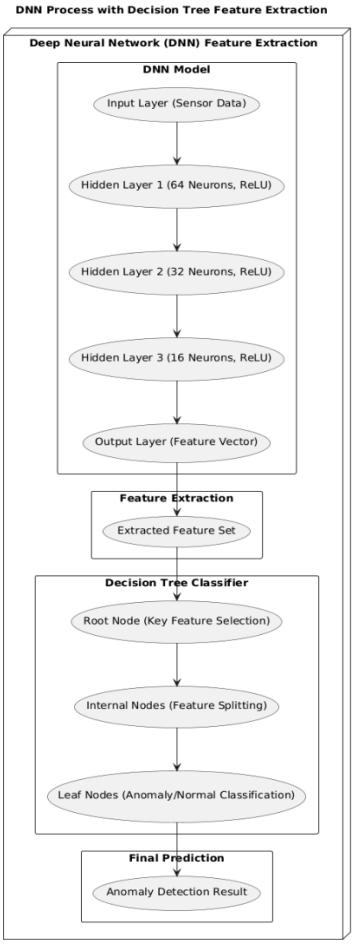

Fig. 3.2: Proposed DNN with DTC model.

However, in the hybrid setup, the final softmax layer is omitted during feature extraction. Instead, the outputs from the penultimate layer (the

16- neuron layer) or even the combination of all hidden layers up to—but not including—the final classification layer are used as the new feature representation. These features are considered robust and discriminative because they are learned automatically from the raw data through multiple levels of abstraction.

Decision Tree Classifier (Ensemble Classifier): Once the DNN has been trained, the intermediate feature representations are extracted and used to train a Decision Tree Classifier. DTC is an ensemble method that constructs multiple decision trees during training and outputs the class that is the mode of the classes (classification) of the individual trees. This step helps to mitigate overfitting and improves the robustness of the final predictions, particularly in scenarios where the data might be noisy or imbalanced.

Layer Architecture Summary:

1. **Input Layer:**
   - Receives the feature vector from the preprocessed inertial sensor data.
2. **Hidden Layers (Feature Extraction):**
   - **Dense Layer 1:** 128 neurons, ReLU activation.
   - **Dense Layer 2:** 64 neurons, ReLU activation.
   - **Dense Layer 3:** 32 neurons, ReLU activation.
   - **Dense Layer 4:** 16 neurons, ReLU activation.
3. **Output Layer (for standalone DNN classification):**

Dense Layer 5: 8 neurons, softmax activation (used during DNN training, but excluded when extracting features for the RFC)

. 3.2.2 Software Requirements

Python 3.7.6

Python 3.7.6 serves as a pivotal version for developers and researchers due to its robust features, backward compatibility, and widespread support across a variety of libraries and frameworks. Released during a time when machine learning and data science tools were rapidly evolving, Python 3.7.6 provided a stable and consistent platform. This version includes critical improvements like enhanced asyncio functionality for asynchronous programming, increased precision for floating-point numbers, and optimized data structures. It became the go-to version for compatibility with popular libraries like TensorFlow 2.0, PyTorch, and Pandas, ensuring seamless integration and efficient execution for both academic and industrial applications.

Compared to older Python versions, 3.7.6 introduced several features such as dataclasses, which simplified boilerplate code for object-oriented programming. The improved async and await syntax made concurrent programming more intuitive, while changes to the standard library enhanced usability and performance. Over newer versions, Python 3.7.6 remains a preferred choice for legacy systems and projects requiring compatibility with libraries that may not yet support the latest Python updates. Its combination of stability and maturity ensures that it is reliable for long-term projects, especsially in environments where upgrading the Python interpreter might disrupt existing workflows.

Packages

python -m pip install --upgrade pip
pip install Cython
pip install tensorflow==1.14.0
pip install keras==2.3.1
pip install pandas==0.25.3
pip install scikit-learn==0.22.2.post1
pip install imutils
pip install matplotlib==3.1.1
pip install opencv-python==4.8.0.74
pip install seaborn==0.10.1
pip install h5py==2.10.0
pip install numpy==1.19.2
pip install jupyter
pip install protobuf==3.20.*
pip install scikit-image==0.16.0

TensorFlow Environment

TensorFlow provides a comprehensive ecosystem for building, training, and deploying machine learning models. Its support for numerical computation and deep learning applications makes it a staple in AI research and development. By offering a flexible architecture, TensorFlow enables deployment across a variety of platforms, including desktops, mobile devices, and the cloud. The ability to scale across CPUs, GPUs, and TPUs ensures that TensorFlow is suitable for both small experiments and large-scale production systems. TensorFlow's transition from older versions, like 1.x, to 2.x brought significant improvements in ease of use, including the introduction of the tf.keras API for building models, eager execution for dynamic computation, and enhanced debugging capabilities. Compared to newer frameworks, TensorFlow retains a strong advantage due to its mature community support, extensive documentation, and integration with TensorFlow Extended (TFX) for managing production pipelines. Its compatibility with other libraries and tools, such as Keras and TensorBoard, makes it a robust choice for end-to-end machine learning solutions.

4. CONCLUSION

The project successfully implements an anomaly detection system for industrial control systems (ICS) using the HAI Security dataset. The system utilizes machine learning (Naïve Bayes) and deep learning (DNN with Decision Tree) to classify network behavior as normal or under attack. Preprocessing steps, including timestamp conversion, feature extraction, and standardization, enhance model performance. The results demonstrate that deep learning models can effectively detect anomalies, but challenges such as data imbalance and feature relevance impact accuracy. The project highlights the potential of AI-driven security

monitoring in industrial settings, offering a proactive approach to cybersecurity threats. In the future, this anomaly detection system can be enhanced by integrating real-time streaming analysis for immediate threat detection. Advanced techniques like hybrid deep learning models (e.g., CNN-LSTM) and reinforcement learning could improve classification accuracy. Additionally, incorporating explainable AI (XAI) can help in understanding model decisions, making it more reliable for industrial applications. Further research can focus on transfer learning to generalize detection across different ICS environments. Deploying this system in edge computing devices will allow for low-latency, real-time threat monitoring, enhancing the security of industrial automation systems.

5. REFERENCES

[1]. Putchala, M.K. Deep Learning Approach for Intrusion Detection System (ids) in the Internet of Things (iot) Network Using Gated Recurrent Neural Networks (gru). Master's Thesis, Wright State University, Dayton, OH, USA, 2020.

[2]. Du, Y.; Huang, Y.; Wan, G.; He, P. Deep Learning-Based Cyber–Physical Feature Fusion for Anomaly Detection in Industrial Control Systems. *Mathematics* **2022**, *10*, 4373.

[3]. Goh, J.; Adepu, S.; Tan, M.; Lee, Z.S. Anomaly detection in cyber-physical systems using recurrent neural networks. In Proceedings of the 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), Singapore, 12–14 January 2020; pp. 140–145.

[4]. Mokhtari, S.; Abbaspour, A.; Yen, K.K.; Sargolzaei, A. A machine learning approach for anomaly detection in industrial control systems based on measurement data. *Electronics* **2021**, *10*, 407.

[5]. Wolsing, K.; Thiemt, L.; Sloun, C.V.; Wagner, E.; Wehrle, K.; Henze, M. Can industrial intrusion detection be simple? In Proceedings of the European Symposium on Research in Computer Security, Copenhagen, Denmark, 26–30 September 2022; pp. 574–594.

[6]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in 2020 IEEE symposium on computational intelligence for security and defense applications, pp. 1–6.

[7]. Á. L. P. Gómez, L. F. Maimó, A. H. Celdrán, F. J. G. Clemente, C. C. Sarmiento, C. J. D. C. Masa, and R. M. Nistal, "On the generation of anomaly detection datasets in industrial control systems," IEEE Access, vol. 7, pp. 177460–177473, 201.

[8]. L. Faramondi, F. Flammini, S. Guarino, and R. Setola, "A hardware-in-the- water distribution testbed dataset for cyber-physical security testing," IEEE Access, vol. 9, pp. 122385–122396, 2021.

[9]. M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge- iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," IEEE Access, vol. 10, pp. 40281–40306, 2022.

[10]. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," Ieee Access, vol. 8, pp. 165130–165150, 2020.

[11]. Ozay, M.; Esnaola, I.; Vural, F.T.Y.; Kulkarni, S.R.; Poor, H.V. Machine learning methods for attack detection in the smart grid. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *27*, 1773–1786.

[12]. Pan, S.; Morris, T.; Adhikari, U. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Trans. Smart Grid* **2021**, *6*, 3104–3113.

[13]. Choi, S. HIL-Based Augmented ICS (HAI) Security Dataset. 2020. Available online: **https://github.com/icsdataset/hai**.