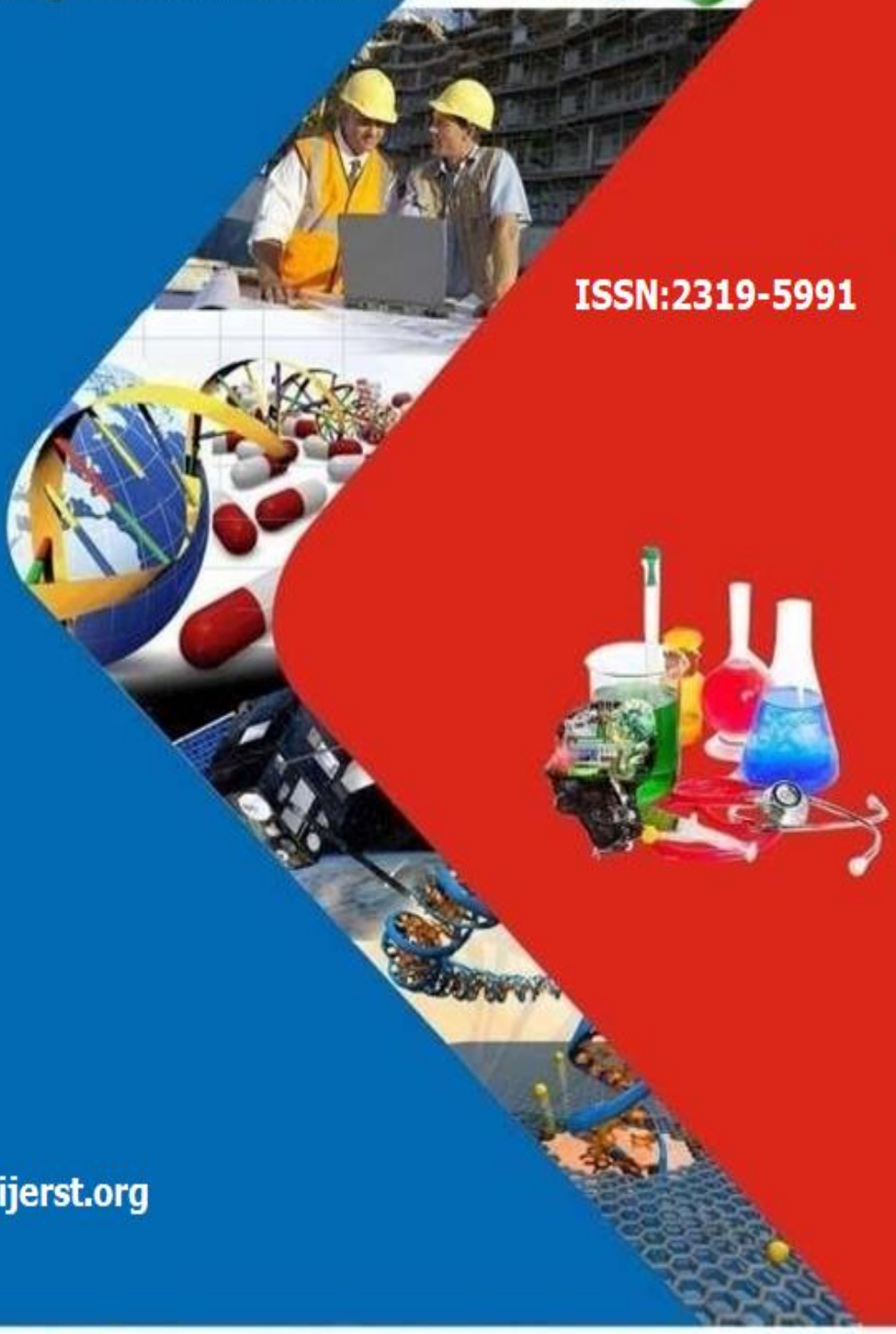
International Journal of Engineering Research and Science & Technology

IJERST

ISSN:2319-5991



www.ijerst.org

E-mail: editor@ijerst.org or ijerst.editor@gmail.com

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

ANOMALY DETECTION AND ATTACK CLASSIFICATION FOR TRAIN REAL-TIME ETHERNET

MS.M.ANITHA[1], MR.K.HAREESH [2], E.SIRISHA[3]

[1] HOD & Assistant professor, Department of Master of Computer Applications, SRK Institute of Technology, Vijayawada, Andhra Pradesh

[2] Assistant professor, Department of Master of Computer Applications, SRK Institute of Technology, Vijayawada, Andhra Pradesh

[3] MCA Student, Department of Master of Computer Applications, SRK Institute of Technology, Vijayawada, Andhra Pradesh

ABSTRACT

Cyberattacks and abnormalities are becoming more common in current rail networks due to their increasing dependence on real-time Ethernet-based connectivity. Manual monitoring solutions are not up to the task of detecting these threats. Manual solutions sometimes use static thresholds and established rules to detect suspicious traffic, which may lead to slower reaction times, human mistake, and an inability to keep up with changing attack trends. Because of these restrictions, sophisticated and automated intrusion detection systems are essential. The goal of this project is to create a system that uses machine learning to identify cyberattacks and abnormalities in railway communication networks by analyzing real-time data. Using a Tkinter-based graphical user interface, users may import, prepare, and analyze datasets using a variety of classifiers, such as Support Vector Machines (SVM), Random Forest, Decision Trees, and sophisticated optimization methods like Genetic Algorithms (GA) and Particle Swarm Optimization (3SO). Selecting features using SFS, GA, and PSO improves model performance while reducing dimensionality. The system is taught to identify harmful traffic, such port scans and denial-of-service assaults, using datasets like KDD99. These datasets include information on network behaviors and characteristics, such as protocol type, service, and flag status. Compared to more conventional approaches, SVM with GA and PSO achieves an accuracy of up to 99% in experiments. The paramount need of protecting transport facilities, the compromise of which might have disastrous effects, is the driving force behind this initiative. By incorporating smart algorithms that can adjust to new dangers, the suggested system provides an automated, scalable, and powerful substitute for human detection techniques. It has an easy-to-understand UI that makes it great for usage in real-world settings, and it speeds up and enhances the accuracy of detection. In the end, this study enhances cyber resilience in railway networks by addressing the limits of human intrusion detection methods and proposing an effective ML-driven alternative.

Keywords: SVM, Random Forest, Decision Trees, Sophisticated optimization, Genetic Algorithms.

1. INTRODUCTION

1. Overview

Train Real-Time Ethernet (TRTE) allows for smooth and time-critical communication between onboard and wayside devices, which has been adopted due to the rising

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

digitization of railway operations. Stopping, signaling, train control, and PIS are all vital operations that TRTE networks provide. The interconnection of railways increases the infrastructure's safety, dependability, and efficiency, but it also makes it more susceptible to cyber-attacks and operational irregularities.

The main objective of this project is to apply machine learning (ML) methods to identify suspicious behavior and categorize any cyber-attacks on TRTE networks. Attacks like replay, spoofing, and denial-of-service (DoS) may put operations and communication at risk. Intelligent algorithms such as K-Nearest Neighbours (KNN) and Naive Bayes may be used to monitor network activity in real-time, detect abnormalities, and categorize potential dangers.

It is of the utmost importance to guarantee the security of TRTE systems while the Indian Railways undertake massive digital transformation programmed like Kavach and SMART Railways. In addition to bolstering predictive maintenance and risk reduction, an efficient anomaly detection and categorization system enhances operational continuity and safety. Findings from this study will help in the development of smart railway networks that are both robust and capable of detecting and responding to hostile actions on their own.

2. Research Motivation

Train Real-Time Ethernet (TRTE) is an essential component for current train systems that depend on real-time communication. It allows for secure, efficient, and timely data interchange. Cybercriminals and data abnormalities increasingly target more sophisticated and interconnected systems. Critical communication between control units may be disrupted by attacks such data injection, spoofing, and Denial-of-Service (DoS), which can cause delays, operational failures, or accidents.

This project is driven by a desire to improve railway operations' safety and dependability by using machine learning for attack categorization and real-time anomaly identification. The suggested solution seeks to make the railway ecosystem safer and smarter by reducing risks, preventing security breaches, and ensuring continuous train communication via the proactive identification of anomalous patterns and harmful behavior in TRTE networks.

3. Problem Definition

There have been both new opportunities for efficient communication and greater susceptibility to cyber-attacks and system abnormalities brought about by the fast digital transformation of railway systems, particularly with the deployment of Train Real-Time Ethernet (TRTE). In real-time, these networks transfer vital monitoring and control data across different subsystems; any interruption, whether caused by bugs or hostile assaults, may have a devastating effect on operational dependability and safety. The current security measures, which are often signature-based or static, are unable to identify the complex and ever-changing attacks that aim to sneak up on these systems.

Unfortunately, there aren't any smart and adaptable systems that can identify anomalies in real-time and correctly categorize cyber-attack types in these networks, even though TRTE is vital to current railway infrastructure. Because TRTE traffic is so complex and moves at such a fast pace, it requires strong machine learning models that can learn from both past and present data to differentiate between safe and dangerous patterns. Filling this void is critical for smart railway communication system resilience, quick reaction, and proactive threat mitigation.

4. Significance

In the era of smart transportation, Train Real-Time Ethernet (TRTE) systems have become

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

essential for enabling fast, reliable, and synchronized communication among various train control units. However, with this advancement comes an increased risk of cyber-attacks and system anomalies that can compromise safety, reliability, and operational efficiency. This study focuses on developing intelligent solutions for anomaly detection and attack classification to ensure the secure and smooth functioning of railway networks.

1) Enhanced Security: By identifying and classifying potential threats in real-time, the system can help prevent cyber-attacks that could disrupt critical train operations.
2) Improved Safety: Early detection of anomalies reduces the risk of accidents caused by faulty communication or system failures in the train's control network.
3) Operational Continuity: Ensures that even in the face of potential threats, timely responses can keep railway systems running with minimal interruptions.
4) Data-Driven Decision Making: Machine learning-based insights provide railway authorities with valuable data to improve infrastructure resilience and incident response.
5) Support for Smart Infrastructure: The research aligns with the vision of digital India and smart transportation systems, contributing to the modernization and sustainability of the railway network.

Applications

1) Intrusion Detection Systems (IDS): Automatically monitor TRTE traffic to detect unauthorized access or malicious behavior in real time.
2) Predictive Maintenance: Identify unusual patterns in network communication that may signal hardware faults or system degradation before failures occur.
3) Network Traffic Optimization: Analyze normal vs. abnormal data flows to enhance bandwidth usage and reduce latency in communication systems.
4) Automated Emergency Response: Trigger predefined safety protocols upon detection of critical anomalies or cyber-attacks to ensure passenger safety.
5) Cybersecurity Policy Enforcement: Support dynamic rule-based access controls and firewall management by classifying types of attacks accurately.

2. LITERATURE REVIEW

The increasing complexity of cyber threats and vulnerabilities necessitates robust detection mechanisms. Machine Learning (ML) algorithms have been widely explored to develop Intrusion Detection Systems (IDS) to identify potential cyber threats. This literature survey covers various studies that focus on cybersecurity threats, intrusion detection mechanisms, risk assessment, and the application of ML algorithms in IDS.

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

Humayun et al. (2020) conducted a systematic mapping study on cybersecurity threats and vulnerabilities, categorizing different security issues that affect systems. Cherdantseva et al. (2016) provided an extensive review of cybersecurity risk assessment techniques for SCADA systems, emphasizing their weaknesses. Zhang (2016) analyzed the safety risks associated with Industrial Control Systems (ICS), identifying potential weaknesses that could be exploited by cyber threats. Xue (2016) presented an overview of intrusion detection technology, providing a historical perspective on the evolution of IDS.

Anderson (1980) and Anderson et al. (1995) introduced some of the foundational works in intrusion detection, emphasizing behavior-based detection. Ramanathan (2002) proposed WADeS, a tool for detecting Distributed Denial of Service (DDoS) attacks. Zhou et al. (2017) designed anomaly-based intrusion detection models to improve security in industrial process automation. Chen et al. (2019) proposed an online detection method for data injection attacks in smart grids using novel ML techniques.

Marton et al. (2013) explored data-driven methods for condition monitoring and maintenance, which contributed to ML-based anomaly detection models. Adepu and Mathur (2018) introduced distributed attack detection techniques in a water treatment plant. Wu and Moon (2019) developed an IDS for cyber-manufacturing systems. Zhao (2013) proposed an abnormal detection algorithm for ICS environments, enhancing real-time detection capabilities.

Das et al. (2019) demonstrated how ensemble ML techniques improve DDoS detection performance. Liu et al. (2012) introduced isolation-based anomaly detection techniques to enhance intrusion detection systems. Thaseen et al. (2019, 2017) proposed an IDS model using chi-square feature selection and SVM classifiers for improving intrusion detection accuracy.

Mirjalili and Lewis (2016) presented the Whale Optimization Algorithm, which has been applied to optimize intrusion detection models.

Esquivel and Esquivel (2009) investigated TCP fingerprinting techniques for spam filtering. Kim et al. (2017) developed a deep neural network-based intrusion detection method. Wang et al. (2014) presented research on support vector mechanisms, highlighting their application in IDS. Kok et al. (2020) evaluated the effectiveness of machine learning in detecting ransomware attacks. Platt (1999) introduced probabilistic outputs for support vector machines (SVMs), improving classification accuracy.

Rasmussen (2003) discussed Gaussian Processes in ML, which have been applied to intrusion detection systems. Zhou (2016) provided insights into ML algorithms and their applications in cybersecurity. The KDD Cup 99 dataset (available online) is widely used for training IDS models. Aburomman and Reaz (2016)

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

proposed an SVM-kNN-PSO ensemble method for improving intrusion detection accuracy. Feng et al. (2014) combined SVMs with ant colony optimization networks for network intrusion detection.

Manzoor and Kumar (2017) presented a feature reduction approach using artificial neural networks (ANN) for IDS. Casas et al. (2012) explored unsupervised learning methods to detect unknown intrusion patterns without prior knowledge, improving adaptive intrusion detection capabilities.

3. PROPOSED SYSTEMS

1 Overview

The proposed system focuses on implementing a machine learning-based solution for real-time anomaly detection and attack classification within Train Real-Time Ethernet (TRTE) networks. It utilizes data-driven models to monitor network traffic, identify deviations from normal behavior, and classify various types of cyber threats. The system aims to enhance cybersecurity and operational reliability by integrating lightweight and accurate algorithms like K-Nearest Neighbors (KNN), enabling quick detection and response with minimal latency.

Step 1: Dataset Selection – KDD99

The KDD99 dataset has been chosen for this research as it is one of the most widely used datasets for anomaly detection and cyberattack classification. This dataset consists of network traffic data captured from a simulated environment, including both normal and attack instances. The dataset contains a total of 41 features, which include basic network connection attributes, content-based features, and traffic-related statistical characteristics. Attacks in KDD99 are categorized into four major types: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L). Since real-time Ethernet in railway communication networks also faces similar types of cyber threats, KDD99 serves as an ideal dataset for training and evaluating machine learning models in anomaly detection and attack classification.

Step 2: Dataset Preprocessing – Null Value Removal and Label Encoding

Raw datasets often contain missing values, duplicate entries, and inconsistent formats, which must be cleaned before applying machine learning models. In this step, null values are removed to ensure that incomplete records do not introduce bias in model training. Next, categorical data is transformed into numerical format using label encoding. In KDD99, categorical features such as protocol type (TCP, UDP, ICMP) and attack labels need to be converted into numeric representations to be effectively processed by machine learning algorithms. Additionally, feature scaling techniques like Min-Max Scaling or Standardization are applied to normalize the data distribution, ensuring that all features contribute equally during model training. This preprocessing step significantly improves the efficiency and accuracy of the models used in later stages.

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

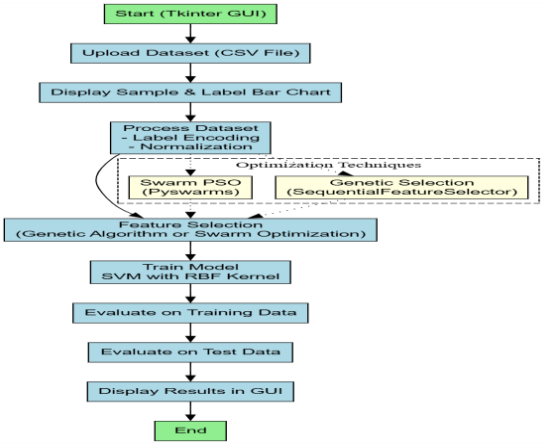


Figure 4.1: Block diagram

Step 3: Existing Model Building – SVM with PSO, SVM, Random Forest, CART Algorithm

In the existing system, multiple machine learning models are implemented and compared to determine their effectiveness in detecting network anomalies. The Support Vector Machine (SVM) with Particle Swarm Optimization (PSO) is one of the key models used. SVM is a powerful classification algorithm that works by finding the optimal hyperplane to separate different classes of data. However, its performance highly depends on parameter selection,

which is why PSO is integrated to optimize SVM parameters automatically. Additionally, Random Forest and CART (Classification and Regression Tree) algorithms are used as benchmark models.

1. **Random Forest** is an ensemble learning method that constructs multiple decision trees and combines their predictions to improve classification accuracy. It is highly effective in handling large datasets with high-dimensional features.

2. **CART (Classification and Regression Trees)** is a decision tree-based algorithm that splits data into hierarchical structures based on feature importance, making it useful for rule-based anomaly detection.

These models are trained and evaluated on the preprocessed KDD99 dataset to analyze their performance in detecting cyber threats in train real-time Ethernet networks.

Step 4: Proposed Model Building – SVM with Genetic Algorithm (GA)

To improve the limitations observed in existing models, this research proposes the Support Vector Machine (SVM) integrated with Genetic Algorithm (GA). Unlike traditional SVM, which relies on manually tuned hyperparameters, GA optimizes the selection of SVM parameters by mimicking the process of natural selection. GA works by generating multiple candidate solutions (chromosomes), evaluating their fitness, applying selection, crossover, and mutation operations to evolve the best hyperparameters for SVM.

By using GA for feature selection and parameter optimization, the SVM classifier achieves higher accuracy, reduced false positives, and better generalization to new attack patterns. This adaptive learning capability of GA ensures that the model effectively detects zero-day attacks and novel cyber threats in railway networks. The SVM-GA model is trained and validated using the same preprocessed

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

KDD99 dataset, and its results are compared against existing models.

Step 5: Performance Comparison of Existing and Proposed Algorithms

To evaluate the effectiveness of the proposed SVM-GA model, a performance comparison is conducted against the existing models (SVM with PSO, Random Forest, and CART). Several performance metrics are used, including:

1. **Accuracy** – Measures the overall correctness of classification.
2. **Precision** – Indicates how many of the predicted anomalies were actually attacks.
3. **Recall** – Determines the model's ability to detect actual attacks.
4. **F1-score** – A balance between precision and recall.
5. **False Positive Rate (FPR)** – Indicates the proportion of normal traffic misclassified as attacks.

4.2 Dataset Preprocessing

Data splitting and preprocessing are critical steps in this research to ensure that the machine learning model efficiently learns from the dataset and generalizes well on unseen data. The dataset used in this study, such as KDD99, contains both normal and anomalous (attack) network traffic. Various preprocessing techniques are applied to refine the dataset before feeding it into the model.

Data Splitting

The dataset is divided into two parts using an **80-20 split**:

1. **80% for training**: The model learns patterns from this data.
2. **20% for testing**: The model is evaluated on this unseen data.

A proper train-test split prevents overfitting and ensures that the model generalizes well.

Data Preprocessing Steps

1. **Handling Missing Values:** Any missing values in the dataset are filled using statistical methods like mean, median, or mode.
2. **Label Encoding:** Categorical features such as protocol type, service, and flag are converted into numerical values.
3. **Feature Scaling & Normalization:** Feature values are scaled between 0 and 1 to remove bias due to varying numerical ranges.
4. **Feature Selection:** Techniques like Principal Component Analysis (PCA) or Particle Swarm Optimization (PSO) are applied to remove irrelevant features, reducing computational complexity.

These preprocessing techniques ensure the dataset is optimized for model training, making the learning process more efficient and accurate.

4.3 EDA

Exploratory Data Analysis (EDA) is a crucial phase in data analysis that enables the researcher to gain insights into the dataset. It helps identify patterns, detect anomalies, and understand relationships between features. In the context of *Train Real-*

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

Time Ethernet (TRTE) networks, EDA is pivotal in preparing the data for anomaly detection and attack classification, which ensures the creation of reliable machine learning models. The EDA process involves cleaning the data, visualizing the distributions, and understanding the structure of the data before any machine learning models are applied.

Step 1: Data Collection & Understanding

The first step in EDA is to load and understand the dataset. This involves checking the structure of the dataset, the number of rows and columns, and the types of features (numerical, categorical, or temporal). It is important to examine initial observations, such as the presence of attack labels, the timestamped entries, and any other relevant fields. Understanding the dataset's composition allows researchers to better assess its complexity and the potential challenges in the analysis, such as the imbalance between attack and non-attack samples.

Step 2: Data Cleaning

In the data cleaning phase, any missing values or inconsistencies need to be addressed. This could involve filling missing values with statistical imputation methods like the mean, median, or mode for numerical features, or removing rows with missing data if necessary. Duplicates should also be detected and removed to ensure that the dataset is accurate. Additionally, ensuring consistency in data formatting, such as standardizing time formats or correcting categorical values, is an important part of cleaning.

Step 3: Descriptive Statistics

Once the data is clean, it's time to generate summary statistics. Descriptive statistics, including measures like the mean, median, mode, standard deviation, minimum, and maximum, are calculated for numerical features to understand the central tendency and spread of the data. For categorical features, frequencies or counts are computed to understand the distribution of each category. Identifying outliers is also crucial at this stage, as they can distort analysis and modeling. Methods like the Interquartile Range (IQR) or Z-score can help in detecting and managing outliers.

Step 4: Data Visualization

Data visualization is a powerful tool for EDA, as it allows for a clear visual understanding of the data's distribution and relationships. For numerical data, histograms, boxplots, and density plots are helpful to visualize the distribution, spread, and outliers. For categorical data, bar charts or pie charts help in understanding the frequency of different categories, such as attack types or protocol types. Additionally, correlation heatmaps are useful in identifying the relationships between different features, allowing for the discovery of strong associations or potential multicollinearity between features.

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

Step 5: Feature Engineering

Feature engineering is an essential step to improve the predictive power of machine learning models. In this step, new features may be derived from existing ones. For example, creating new features like the average packet size per second or the flow duration can provide more informative inputs for the model. Additionally, categorical variables are often encoded numerically using methods such as One-Hot Encoding or Label Encoding. Irrelevant features or redundant columns are also removed during this step to streamline the dataset and reduce noise.

Step 6: Class Imbalance Handling

Class imbalance is a common problem in anomaly detection datasets, where normal traffic vastly outnumbers attack traffic. This step involves analyzing the distribution of attack and non-attack labels. If there is a significant imbalance, techniques such as oversampling (e.g., SMOTE) or undersampling can be used to balance the dataset. Balancing the classes ensures that the model is not biased toward predicting the majority class and can improve its ability to correctly classify minority class instances, such as specific **types** of attacks.

4.4 Model Building and Training

4.4.1 SVM with PSO (Particle Swarm Optimization)

Support Vector Machine (SVM) is a widely used supervised learning method for classification problems. It performs well in high-dimensional spaces and is effective in cases where the number of dimensions exceeds the number of samples. However, its performance is highly dependent on selecting the optimal hyperparameters. To overcome the limitations of manual tuning, Particle Swarm Optimization (PSO) is used. PSO is a metaheuristic inspired by the social behavior of birds and fish, capable of efficiently exploring complex search spaces. By combining SVM with PSO, we can automate hyperparameter selection, thereby improving classification accuracy and reducing computation time. This hybrid model is particularly well-suited for detecting anomalies and classifying attacks in critical systems like Train Real-Time Ethernet (TRTE).

Step 1: Data Preprocessing

In the first step, the dataset is preprocessed to make it suitable for classification. This includes handling missing values by either imputation or removal, and normalizing the feature values to ensure all features contribute equally to the SVM model. Normalization typically uses techniques like Min-Max scaling or Z-score standardization. Additionally, if the dataset contains categorical variables, they are encoded into numerical form using methods such as one-hot encoding or label encoding. Preprocessing ensures data consistency and prepares the features for optimal learning.

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

Step 2: Initialization of PSO Parameters

After preprocessing, the next step involves initializing the PSO algorithm. This includes defining a population of particles, where each particle represents a potential solution consisting of SVM hyperparameters commonly the penalty parameter C and kernel parameter gamma. Each particle is initialized with a random position (solution) and a velocity vector that determines its movement through the solution space. Other parameters like the number of particles, maximum iterations, and inertia weight are also set at this stage.

Step 3: Fitness Evaluation of Particles

Each particle's position (i.e., set of SVM parameters) is evaluated using a fitness function. The fitness function measures how well the SVM model performs with the given parameters, typically using accuracy, F1-score, or a custom metric on a validation set. The performance result is stored as the particle's personal best (pBest), and the best-performing solution among all particles is identified as the global best (gBest). This evaluation guides the future movements of the particles.

Step 4: Particle Position and Velocity Update

In this step, the position and velocity of each particle are updated. The new velocity is calculated based on the particle's previous velocity, the distance from its personal best position, and the distance from the global best position. This update simulates both individual learning and social influence. The position is then updated using the new velocity. This iterative process enables the swarm to explore and exploit the search space to find better solutions with each generation.

Step 5: SVM Training with Optimal Parameters

Once the PSO algorithm converges either by reaching a maximum number of iterations or satisfying a convergence condition the optimal set of hyperparameters (C and gamma) is selected from the global best particle. These optimized parameters are then used to train the final SVM model on the training data. This trained model is expected to deliver better generalization and accuracy due to its fine-tuned hyperparameters.

Step 6: Testing and Evaluation

In the final step, the trained SVM model is evaluated on unseen test data to measure its effectiveness. Key performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix are computed to assess the model's classification ability. The results are compared against baseline models or non-optimized SVM to demonstrate the improvement achieved by using PSO for parameter tuning.

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

5. RESULTS

Figure 1: TKinter GUI

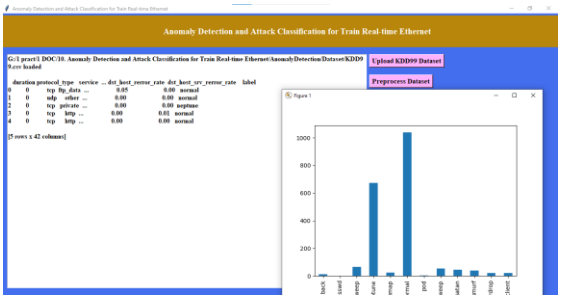


Figure 2: Uploaded the Dataset

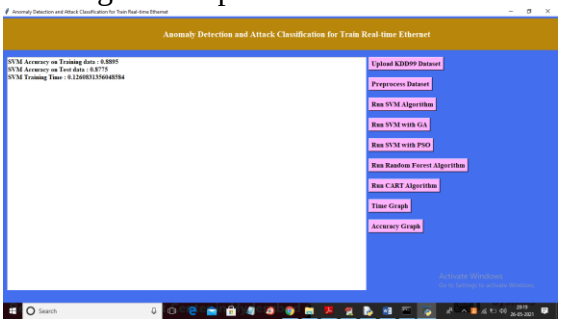


Figure 3: Metrics of SVM

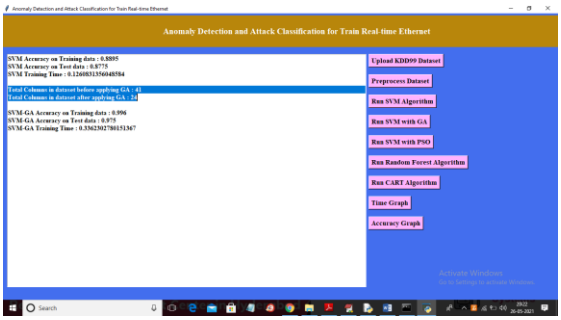


Figure 4: Metrices of SVM with GA

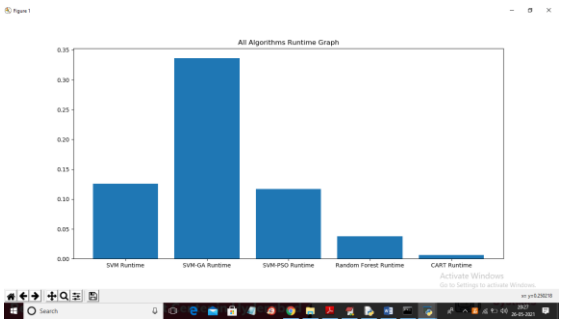


Figure 7: shows count plot

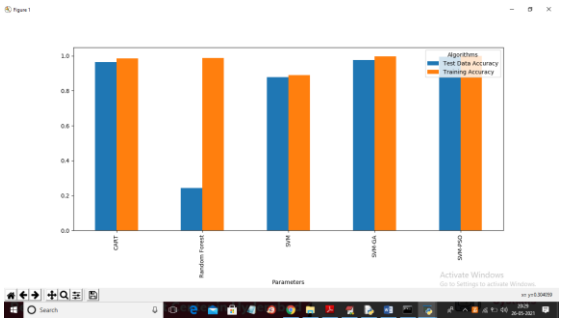


Figure 8: comparison graph

6. CONCLUSION

This project presents a robust, intelligent, and automated approach for anomaly detection and cyberattack classification in real-time Ethernet-based railway communication networks. Leveraging machine learning classifiers like SVM, Random Forest, and Decision Trees, along with advanced feature selection and optimization techniques such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), the system significantly improves detection accuracy and efficiency. The use of GUI-based interaction through Tkinter enhances usability, making the system accessible for operational environments. Experimental results show that integrating optimization algorithms with machine learning models leads to a considerable boost in performance. Specifically, SVM combined with PSO achieved a remarkable 99% accuracy, demonstrating its potential for real-world deployment. The system not only detects

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

known threats like DoS attacks and port scans but also adapts to evolving patterns in network behavior. This makes it a scalable and future-proof solution for enhancing the cybersecurity posture of railway infrastructure.

REFERENCES

1. M. Humayun, M. Niazi, N. Z. Jhanjhi, M. Alshayeb and S. Mahmood, "Cyber security threats and vulnerabilities: A systematic mapping study", Arabian J. Sci. Eng., vol. 45, no. 3, pp. 19, Apr. 2020.
2. Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, et al., "A review of cyber security risk assessment methods for SCADA systems", Comput. Secur., vol. 56, no. 1, pp. 1-27, Feb. 2016.
3. S. Zhang, "Safety status and risk analysis of industrial control systems–one of the safety risk analyses of ICS industrial control systems", Netw. Comput. Secur., vol. 1, no. 5, pp. 15-19, Jun. 2016.
4. J. F. Xue, Intrusion Detection Technology, Beijing, China:Posts and Telecom Press, 2016.
5. J. P. Anderson, "Computer security threat monitoring and surveillance", 1980.
6. D. Anderson, T. Frivold and A. Valdes, Next-Generation Intrusion Detection Expert System (NIDES): A Summary, 1995.
7. A. W. Ramanathan, "WADeS: A tool for distributed denial of service attack detection", 2002.
8. C. J. Zhou, S. Huang, N. Xiong, S. H. Yang, H. Li, Y. Qin, et al., "Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation", IEEE Trans. Syst. Man Cybern. Syst., vol. 45, no. 10, pp. 2216-2468, 2017.
9. R. Chen, X. Li, H. Zhong and M. Fei, "A novel online detection method of data injection attack against dynamic state estimation in smart grid", Neurocomputing, vol. 344, pp. 73-81, Jun. 2019.
10. I. Marton, A. I. Sánchez, S. Carlos and S. Martorell, "Application of data driven methods for condition monitoring maintenance", Chem. Eng. Trans., vol. 33, no. 10, pp. 301-306, 2013.
11. S. Adepu and A. P. Mathur, "Distributed attack detection in a water treatment plant: Method and case study", IEEE Trans. Dependable Secure Comput., vol. 16, no. 1, pp. 1-14, Jan./Feb. 2018.
12. M. Wu and Y. B. Moon, "Intrusion detection system for cyber-manufacturing system", J. Manuf. Sci. Eng., vol. 141, no. 3, pp. 7-31, Mar. 2019.
13. H. Zhao, "Research on abnormal detection algorithm of industrial control system", 2013.
14. S. Das, A. M. Mahfouz, D. Venugopal and S. Shiva, "DDoS intrusion detection through machine learning ensemble", Proc. IEEE 19th Int. Conf. Softw. Qual. Rel. Secur. Companion (QRS-C), pp. 471-477, Jul. 2019.
15. F. T. Liu, K. M. Ting and Z.-H. Zhou, "Isolation-based anomaly detection", ACM Trans. Knowl. Discovery Data, vol. 6, no. 1, pp. 1-39, Mar. 2012.
16. I. S. Thaseen, C. A. Kumar and A. Ahmad, "Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers", Arabian J. Sci. Eng., vol. 44, no. 4, pp. 3357-3368, Apr. 2019.
17. I. S. Thaseen and C. A. Kumar, "Intrusion detection model using

International Journal of Engineering Research and Science & Technology
ISSN 2319-5991 www.ijerst.com
Vol. 21, Issue 2, 2025

fusion of chi-square feature selection and multi class SVM", J. King Saud Univ. Comput. Inf. Sci., vol. 29, no. 4, pp. 462-472, Oct. 2017.
18. S. Mirjalili and A. Lewis, "The whale optimization algorithm", Adv. Eng. Softw., vol. 95, no. 5, pp. 16-27, Jan. 2016.
19. H. Esquivel and T. Esquivel, "Router-level spam filtering using tcp fingerprints: Architecture and measurement-based evaluation", Proc. 6th Conf. Email Anti-Spam (CEAS), pp. 1-10, 2009.
20. J. Kim, N. Shin, S. Y. Jo and S. Hyun Kim, "Method of intrusion detection using deep neural network", Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp), pp. 313-316, Feb. 2017.
21. H. Y. Wang, J. H. Li and L. Feng, "Overview of support vector mechanism and algorithm research", Comput. Appl. Res., vol. 31, no. 5, pp. 1281-1286, Dec. 2014.
22. S. H. Kok, A. Azween and N. Jhanjhi, "Evaluation metric for crypto-ransomware detection using machine learning", J. Inf. Secur. Appl., vol. 55, Dec. 2020.
23. J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods", Adv. Large Margin Classifiers, vol. 10, no. 3, pp. 61-74, 1999.
24. C. E. Rasmussen, Gaussian Processes in Machine Learning, vol. 3176, pp. 14, Feb. 2003.
25. Z. H. Zhou, Machine Learning, Beijing, China:Tsinghua Univ. Press, pp. 121-139, 2016.
26. [online] Available: http://kdd.ics.uci.edu/databases/kddcup99/.
27. A. A. Aburomman and M. B. I. Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system", Appl. Soft Comput., vol. 38, pp. 360-372, Jan. 2016.
28. W. Feng, Q. Zhang, G. Hu and J. X. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks", Future Gener. Comput. Syst., vol. 37, pp. 127-140, Jul. 2014.
29. I. Manzoor and N. Kumar, "A feature reduced intrusion detection system using ANN classifier", Expert Syst. Appl., vol. 88, pp. 249-257, Dec. 2017.
30. P. Casas, J. Mazel and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge", Comput. Commun., vol. 35, no. 7, pp. 772-783, Apr. 2012.