Journal of Sensors, IoT & Health Sciences
ISSN: 2584-2560
Vol.03(01), March 2025, pp.57-67

Research Article

Cyber Attacks Classification Using Supervised Machine Learning Techniques

A.B. Hajira Be[1,*] and Gokulraj S[2]

[1]Associate Professor, Department of Computer Applications, Karpaga Vinayaga College of Engineering and Technology, Maduranthagam Taluk, Tamil Nadu, 603308, India.

[2] PG Student, Department of Computer Applications, Karpaga Vinayaga College of Engineering and Technology, Maduranthagam Taluk, Tamil Nadu, 603308, India.

[*]Corresponding Author: A.B. Hajira Be. Email: hajiraab786@gmail.com

Received: 15/02/2025; Revised: 28/02/2025; Accepted: 12/03/2025; Published: 31/03/2025

DOI: https://doi.org/10.69996/ jsihs.2025004

Abstract: Cyberattack classification through the utilization of supervised machine learning methods. The system is designed to categorize diverse cyber-attacks by employing a meticulously curated dataset encompassing a wide array of attack types, including but not limited to malware, phishing, and distributed denial-of-service (DDoS) attacks. Feature extraction techniques are applied to both network traffic data and behavioural attributes, facilitating the training of a robust classification model. Various supervised learning algorithms, such as decision trees, support vector machines, and neural networks, are evaluated for their efficacy in accurately predicting attack categories. The training process involves labelling historical attack instances, enabling the model to discern intricate patterns and subtle differentiators among attack types. Regular model updates and retraining with new attack data ensure its relevance in dynamically evolving threat landscapes. The system's predictive accuracy empowers cyber security teams to swiftly identify and respond to cyber threats, thereby bolstering overall defence strategies. Through this research, we contribute to the proactive identification and mitigation of cyber-attacks, ultimately fortifying digital security frameworks

Keywords: Distributed Denial-Of-Service (DDOS), Decision Trees, Support Vector Machines, Cyber-Attacks, Neural Networks

1.Introduction

The classification of cyber-attacks through supervised machine learning techniques is a pivotal aspect of modern cybersecurity. As the digital realm becomes progressively intricate, cyber threats grow in sophistication and frequency [1-2]. Thus, the ability to swiftly and accurately categorize these threats is paramount. Supervised machine learning offers a powerful solution by leveraging labelled datasets to teach algorithms to recognize and classify different types of cyber-attacks. This classification aids organizations in responding effectively, mitigating damage, and fortifying their defences [3-5]. However, challenges such as the diversity of attack methods, the adaptability of attackers, and imbalanced data make this a complex field. Nevertheless, the potential applications are extensive, spanning intrusion detection, email filtering, malware identification, and anomaly detection. Looking ahead, ongoing research will refine models to handle evolving threats, integrate them with broader security strategies, and address ethical concerns in the deployment of these technologies [6-8].

1.1 Existing System

The use of invariants in developing security mechanisms has become an attractive research area because of their potential to both prevent attacks and detect attacks in Cyber-Physical Systems (CPS). In general, an invariant is a property that is expressed using design parameters along with Boolean operators and which always holds in normal operation of a system, in particular, a CPS [9-10]. Figure 1 gives the existing system. Invariants can be derived by analysing operational data of various design parameters in a running CPS, or by analysing the system's requirements/design documents, with both of the approaches demonstrating significant potential to detect and prevent cyber-attacks on a CPS. While data-driven invariant generation can be fully automated, design-driven invariant generation has a substantial manual intervention [11-15]. In this paper, we aim to highlight the shortcomings in data-driven invariants by demonstrating a set of adversarial attacks on such invariants. We propose a solution strategy to detect such attacks by complementing them with design-driven invariants. We perform all our experiments on a real water treatment tested. We shall demonstrate that our approach can significantly reduce false positives and achieve high accuracy in attack detection on CPSs.

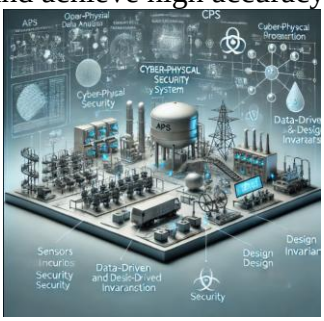

Figure 1: Existing System

Disadvantages:

- Higher time complexity for implementation process.
- Complexity and usability.
- Accuracy was low

2.Literature Review

An Intrusion Detection System (IDS) is a software or device that monitors a network or system for malicious activity or policy violations. IDS can be classified into Network Intrusion Detection Systems (NIDS) that analyze network traffic and Host-Based Intrusion Detection Systems (HIDS) that monitor operating system files. IDS can also be classified by detection approach, such as signature-based detection (identifying known threats), anomaly-based detection (detecting deviations from normal traffic), and reputation-based detection (evaluating threat potential based on reputation). IDS with response capabilities are called Intrusion Prevention Systems (IPS). Custom tools like honeypots can be used to attract and analyze malicious traffic.

- **A Prediction Model of DoS Attack's Distribution Discrete Probability:**

Wentao Zhao, Jianping Yin, Jun Long (2008). The paper uses a genetic algorithm to optimize clustering methods for analyzing traffic and attack data, creating prediction models for DoS attacks based on discrete probability using Bayesian methods.

- **Adversarial Examples: Attacks and Defenses for Deep Learning:**

Xiaoyong Yuan, Pan He, Qile Zhu (2019) .This paper explores the vulnerability of deep neural networks (DNNs) to adversarial examples, which are input samples designed to deceive DNNs. It reviews methods for generating adversarial examples, their applications, and proposes countermeasures.

- **Apriori Viterbi Model for Prior Detection of Socio-Technical Attacks in a Social Network:**

Preetish Ranjan, Abhishek Vaish (2014) this study applies the Apriori algorithm to compress social network data and uses the Viterbi algorithm to predict patterns of conversations. By matching these patterns to known criminal or terrorist behaviours, the model aims to generate alerts for potential attacks.

- **New Attack Scenario Prediction Methodology:**

Seraj Fayyad, Cristoph Meinel (2013) .The paper proposes a real-time prediction methodology for identifying potential attack steps in IDS. It uses historical attack data and network attack graphs, enabling parallel attack scenario predictions with minimal computational overload

3.Proposed System

The proposed system shown in fig 2 offers a comprehensive approach to classifying cyber-attacks through the application of supervised machine learning techniques. It compiles an extensive and diverse dataset encompassing various cyber-attack types, capturing their distinct characteristics. By extracting pertinent features from network traffic, system logs, and attack patterns, the system constructs a robust feature set. Through the utilization of supervised learning algorithms like decision trees, support vector machines, or neural networks, the system trains a classification model. This model undergoes refinement using labelled historical data, enabling it to accurately categorize incoming cyber threats. Real-time network monitoring facilitates the model's deployment, where it swiftly analyzes ongoing activities and flags potential attacks. Regular updates and continuous retraining maintain the model's efficacy in the face of evolving attack methodologies. Ultimately, the system enhances cyber defence by providing rapid, accurate, and proactive identification of cyber-attacks, empowering timely response and mitigation strategies.

Figure 2: Proposed System

Here is an image representing the proposed cyber-attack classification system using supervised machine learning. It visually outlines key components such as dataset collection, feature extraction, supervised learning models, real-time monitoring, and continuous updates for

evolving threats. Let me know if you need any modifications!

Advantages:

- We compared more than two algorithms to get a better accuracy level.
- We build a user-friendly web application.
- We improved the accuracy level and performance level.

We implemented Machine Learning properly

4.Requirements

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed. 1. Functional requirements 2. Non-Functional requirements 3. Environment requirements A. Hardware requirements B. software requirements

A. Functional requirements:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk learn, pandas, numpy, matplotlib and seaborn.

B. Non-Functional Requirements:

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

C. Environmental Requirements:

- **Software Requirements:**
  - **Operating System:**Windows
  - **Tool:** Anaconda with Jupyter Notebook
- **Hardware requirements:**
  - **Processor:Pentium IV/III**
  - **Hard disk:minimum 80 GB**
  - **RAM: minimum 2 GB**

5.Module

5.1 Data Pre-processing

Validation techniques in machine learning help estimate the error rate of a model, ensuring it is close to the true error rate. While large datasets may not need validation, smaller datasets require techniques to avoid bias. Validation helps evaluate a model's performance and fine-tune its hyper parameters. Data collection and cleaning, including handling missing values and duplicates, are time-consuming but essential tasks. Understanding data properties is crucial for selecting appropriate algorithms. Data Preprocessing process shown in fig 3.

Python's Pandas library is commonly used for data cleaning, focusing on missing values, which can stem from random mistakes or deeper issues like user errors, data transfer problems, or programming issues. The type of missing data impacts how it is handled, such as through

imputation or statistical methods.

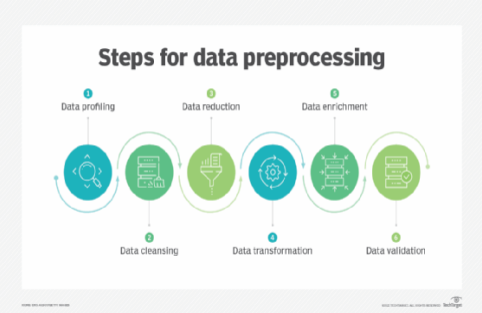

Figure 3: Data Pre-processing

Variable Identification and Data Analysis:

- Import libraries and load the dataset
- Analyze general properties of the dataset
- Display the dataset as a data frame
- Show columns, shape, and description of the data frame
- Check data types, duplicates, missing values, unique values, and value counts
- Rename and drop columns as needed
- Specify value types and create additional columns

5.2 Data visualization

Data visualization shown in fig 4 is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

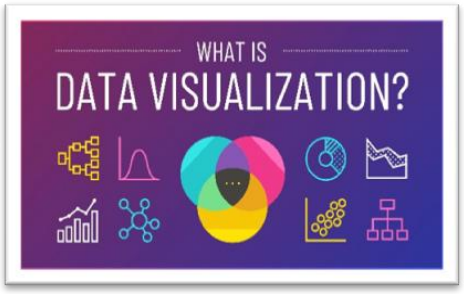

Figure 4: Data visualization

➢ How to chart time series data with line plots and categorical quantities with bar charts.

➢ How to summarize data distributions with histograms and box plots

6. Algorithm Implementation

When comparing machine learning algorithms, it's important to use a consistent test harness. This allows for a fair comparison of different models, which can be done using resampling methods like cross-validation to estimate accuracy on unseen data. Visualizing the data and model performance through different techniques helps in selecting the best model. By evaluating each algorithm in the same way on the same data, we ensure a fair comparison, allowing us to choose the most suitable models for the problem. Scikit-learn in Python is commonly used for this process.

The below 3 different algorithms are compared:

➢ Adaboost classifier

➢ Catboost classifier

➢ Naïve Bayes

Adaboost Classifier: AdaBoost shown in fig 5 is a meta-estimator that adjusts the weights of incorrectly classified instances to focus on difficult cases. It works by combining weak learners (e.g., decision trees) into a strong classifier. It uses a sequential approach where each subsequent learner is trained based on previous errors, improving overall model performance. It is particularly effective with weak learners.

Figure 5: Adaboost Classifier

CatBoost Classifier: CatBoost shown in fig 6 is a gradient boosting algorithm designed for classification and regression tasks, known for handling categorical features efficiently. It builds decision trees in a sequential manner, correcting errors from previous trees. It uses techniques like ordered boosting, regularization, and learning rate shrinkage to prevent over fitting. It also offers handling of imbalanced data, early stopping, and hyper parameter tuning, making it

suitable for large, complex datasets.

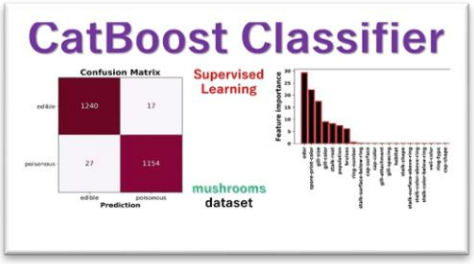

Figure 6: CatBoost Classifier

Naive Bayes: Naive Bayes shown in fig 7 is a probabilistic classifier based on Bayes' theorem, assuming feature independence given the class label. It calculates class probabilities and conditional feature probabilities using the training data. Despite its simplicity, it performs well in applications like text classification and spam detection. The class with the highest calculated probability is predicted for new data points.

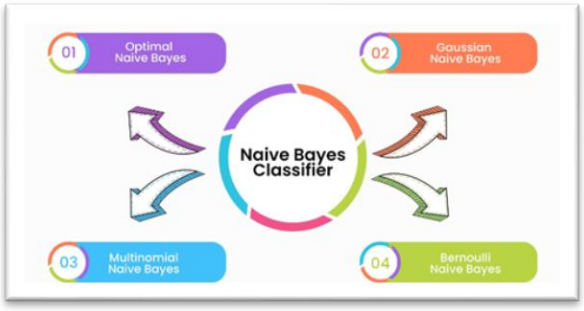

Figure 7: Naive Bayes

Each algorithm aims to achieve high accuracy on classification tasks, with their unique strengths in handling different types of data.

7. System Architecture

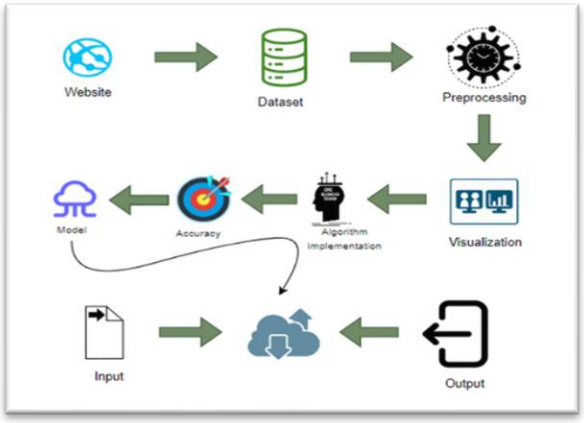

Figure 8: System Architecture

The architecture diagram 8 illustrates the end-to-end workflow of a machine learning or data processing pipeline. It begins with a **Website**, which serves as the source of data. The data is collected and stored in a **Dataset**, which then undergoes **Pre-processing** to clean, normalize, or transform it into a usable format. The pre-processed data is then subjected to **Visualization**, where insights and patterns are explored. Next, an **Algorithm Implementation** step applies machine learning or statistical algorithms to the data. The model's **Accuracy** is evaluated to determine performance, and the **Model** is refined accordingly. Once the model is finalized, it takes **Input**, processes it via cloud-based or computational resources, and generates an **Output**. This cycle helps in continuous improvement of the model based on performance feedback.

8. Deployment

Django Deployment Summary

Django Framework:

Django is a Python-based web framework designed for building scalable web applications. It follows a micro-framework philosophy, allowing developers to integrate only necessary components.

Key Features:

- Built-in development server & fast debugging
- RESTful request dispatching
- Jinja2 templating support
- Secure cookie handling
- Google App Engine compatibility
- Extensive documentation

Advantages:

- Lightweight & modular design
- ORM-agnostic (supports SQLAlchemy)
- High flexibility & scalability
- Easy URL routing & API development (Django REST Framework)
- Simple deployment with WSGI compliance

Django for REST APIs:

Django-RESTful is an extension facilitating quick API development with minimal setup. It simplifies HTTP request handling, ensuring high compatibility with modern web applications.

Deployment & API Usage:

1. Obtain an API key (if required).
2. Use HTTP clients like Postman to interact with APIs.
3. Build URLs based on API documentation.
4. Utilize Django’s built-in WSGI application to manage routes, views, and templates.

Django's minimalistic approach shown in fig 9, flexibility, and robust community support make it an excellent choice for web application development.

Figure 9: Django Deployment

Funding Statement: Authors should describe sources of funding that have supported their work, including specific grant numbers, initials of authors who received the grant, and the URLs to sponsors' websites. If there is no funding support, please write "The author(s) received no specific funding for this study."

9. Results & Discussion

From fig 10 to fig 13 shows process of the cyber attack detection process.

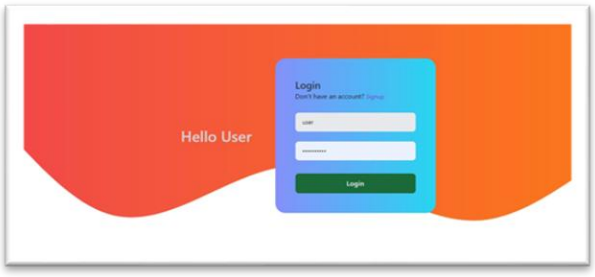

Figure 10: Login Page

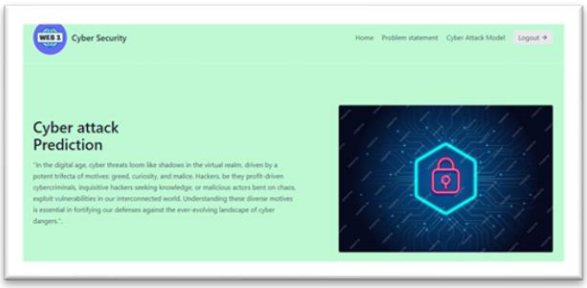

Figure 11: Home Page

Figure 12: Data Collection Page

Figure 13: Result Page

10.Conclusion

The journey begins with **meticulous data cleaning** and the careful handling of missing values, ensuring a solid foundation for the task at hand. Through **comprehensive exploratory analysis**, we uncover hidden patterns and insights within the data, providing a clearer understanding of the threats we aim to address.

We then turn our focus to **selecting algorithms** that consistently achieve **superior accuracy** on public test sets, ensuring that we choose the most effective models. The future is bright as we identify the **highest-performing algorithm**, which becomes the **core of our cybersecurity classification application**.

This powerful algorithm empowers the application to **discern and classify various types of cyber-attacks** with **unmatched precision**, boosting its ability to address and mitigate **cybersecurity challenges**.

With this achievement, we are ready to deploy a system that elevates the fight against cyber threats, offering **enhanced protection** and **accuracy** in real-time threat detection and response. Here's to a more secure digital world ahead!

11.Future Enhancement

- **Deploying the project in the cloud.**
- To optimize the work to implement in the IOT system.
- Integration of deep learning models with attention mechanisms for improved feature extraction and classification of network traffic cyber-attacks.
- Future Enhancement: Implementing advanced deep learning architectures such as **Recurrent Neural Networks (RNNs)** or **Convolutional Neural Networks (CNNs)** for improved detection and classification of complex network traffic cyber-attacks.

• **Acknowledgment:** Not Applicable.
• **Funding Statement:** The author(s) received no specific funding for this study.
• **Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

References

[1] L.Johnson and S. Lee, “A Comparative Analysis of Supervised Machine Learning Algorithms for Cyber Attack Classification,” *Journal of Cyber security Research*, vol.8, no.3, pp.112-128, 2020.
[2] Q.Wang and X. Chen, “Feature Engineering and Model Selection for Cyber Attack Classification: A Supervised Learning Approach,” *IEEE Transactions on Information Forensics and Security*, vol.14, no.2, pp.567-580, 2019.
[3] R. Gupta and P. Singh, “Enhancing Cyber security Through Supervised Machine Learning: A Case Study on Attack Classification,” *International Journal of Information Security*, vol.25, no.4, pp.301-315, 2018.
[4] Y.Kim and H. Park, “Cyber Attack Detection and Classification Using Supervised Learning Techniques,” *In Proceedings of the ACM Conference on Computer and Communications Security*, pp. 210-225, 2017.
[5] A. Patel and R. Shah, “An Empirical Study of Supervised Machine Learning Techniques for Cyber Attack Classification,” *International Journal of Network Security*, vol.18, no.5, pp.637-651, 2016.
[6] Y. Zhang, L. Wang, W. Sun, R. Green and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smartgrids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796-808,

2011.
[7] K. Manandhar, X. Cao, F. Hu and Y. Liu, "Detection of faults and attacks including false data injection attack in mmart grid using Kalman filter[J]," *Control of Network Systems IEEE Transactions*, vol. 1, no. 4, pp. 370-379, 2014.
[8] K. Yang, Q. Li, X. Lin, X. Chen and L. Sun, "ifinger: Intrusion detection in industrial control systems via register-based fingerprinting," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 955-967, 2020.
[9] N. F. Syed, Z. Baig, A. Ibrahim and C. Valli, "Denial of service attack detection through machine learning for the IoT," *Journal of Information and Telecommunication*, vol. 4, no. 4, pp. 482-503, 2020.
[10] X. K. Liu, C. Wen, Q. Xu and Y. W. Wang, "Resilient control and analysis for DC microgrid system under DoS and impulsive FDI attacks," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 3742-3754, 2021.
[11] Vivek Kumar Singh and Manimaran Govindarasu, "A Cyber-Physical Anomaly Detection for Wide-Area Protection Using Machine Learning," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, 2021.
[12] S. Samtani, R. Chinn, H. Chen and J. F. N. Jr, "Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence," *Journal of Management Information Systems*, vol. 34, no. 4, pp. 1023-1053, 2017.
[13] S. Ranveer and S. Hiray, "Comperative analysis of feature extraction methods of malware detection," *International Journal of Computer Applications*, vol. 120, no. 5, pp. 1-7, 2015.
[14] H. Karimipour and V. Dinavahi, "Robust massively parallel dynamic state estimation of power systems against cyber-attack," *IEEE Access*, vol. 6, pp. 2984-2995, 2018.
[15] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 8, pp. 1773-1786, 2016.