International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 07 | July – 2025 DOI: 10.55041/ISJEM04803
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Network Intrusion Detection Using Machine Learning

CH. VASUNDHARA, BOBBADI ADARSH

Assistant Professor, 2MCA Final Semester,

Master of Computer Applications,

Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India

Abstract:

The rapid growth of the internet and communication technologies, there has been a significant increase in network size and the volume of data transmitted. This expansion has led to the emergence of numerous novels cyberattacks, posing serious challenges to network security. Intrusion Detection Systems (IDS) play a vital role in identifying and mitigating such threats by monitoring network traffic to ensure confidentiality, integrity, and availability. Despite ongoing research efforts, IDS still face challenges such as improving detection accuracy, minimizing false alarms, and effectively identifying new types of attacks. To address these issues, recent advancements have focused on integrating machine learning (ML) techniques into IDS. ML-based IDS offer promising solutions for enhancing intrusion detection capabilities by learning from data patterns and adapting to emerging threats. As cyberattacks continue to evolve, the development of intelligent, adaptive IDS using machine learning has become essential for robust network security.

IndexTerms: Machine Learning (ML), Network Intrusion Detection System (IDS), Machine Learning (ML), Random Forest, Data Preprocessing, Python, Matplotlib, Data Mining.

1. INTRODUCTIO

2. For the past few years, network has played a significant role in communication. The computer network allows the computing network devices to exchange information among different systems and individuals. The services of various organizations, companies, colleges, universities are accessed throughout computer network. This leads to a massive growth in networking field. The accessibility of internet has acquired a lot of interest among individuals. In this context, security of information has become a great challenge in this modern area. The information or data that we would like to send is supposed to be secured in such a way that a third party should not take control over them. When we are talking about security, we have to keep three basic factors in our mind: Confidentiality, Integrity and availability.[4] Confidentiality means privacy of information. It gives the formal users the right to access the system via internet. This can be performed suitably along with accountability services in order to identify the authorized individuals. The second key factor is integrity. The integrity service means exactness of information. It allows the users to have self- assurance that the information passed is acceptable and has not been changed by an illegal individual. An Intrusion Detection System (IDS) is used to watch malicious activities over the network. It can sort the unfamiliar records as normal or attack class. First monitoring of the network traffic is done, and then the IDS sorts these networks traffic records into either malicious class or regular class. It acts as an alarm system that reports when an illegal activity is detected. The exactness of the IDS depends upon detection rate

1.1 Existing System

The existing system for intrusion detection uses the K-Nearest Neighbor (KNN) algorithm to classify network traffic as either normal or malicious. It works by monitoring the network traffic within a sub net and comparing it with a known list of attack signatures. When suspicious activity is detected, it triggers an alert to the system administrator. The system includes two types of Intrusion Detection Systems (IDS): Network-based IDS (NIDS) and Host-based IDS (HIDS). NIDS is installed on network devices [6]like routers and monitors incoming and outgoing packets. HIDS runs on individual hosts and monitors file integrity by comparing current files to previously stored versions. [2]If changes are detected, such as file modifications or deletions, it flags a possible intrusion. Although KNN is simple and widely used, it suffers from several limitations. It provides less than 50% accuracy in many cases and is inefficient with large datasets due to high computational cost. It also performs poorly with high-dimensional data, and requires feature scaling to function correctly. Additionally, KNN is highly sensitive to noise, outliers, and missing values. These drawbacks make it less effective in real-world scenarios where data is complex and dynamic. As a result, more efficient and accurate methods are needed for effective intrusion detection.

1.1.1 Challenges:

- **Handling Large Datasets:**

The network traffic data and malware detection datasets are vast, requiring efficient storage, processing, and real-time analysis.

ISSN: 2583-6129
DOI: 10.55041/ISJEM04803

- **Imbalanced Data:**

Most intrusion datasets contain a small number of attack instances compared to normal traffic, which affects model accuracy.

- **Evolving Attack Patterns:**

Caber-attacks evolve over time, so static models may fail to detect new or unknown threats.

- **Real-time Detection Requirement:**

IDS systems must work in real-time, demanding fast and lightweight algorithms with minimal latency.

- **Testing and Validation:**

Ensuring the model works accurately across all test cases, including rare or zero-day attacks, is challenging.

- **Security of the IDS Itself:**

The IDS must be robust against direct attacks that try to disable or bypass it—an often-overlooked challenge in system design.

Proposed system:

The proposed system in the document titled “Network Intrusion Detection Using Machine Learning” introduces a machine learning-based approach to effectively detect and classify malicious activities within a network.[12] To overcome the limitations of traditional detection systems like the K-Nearest Neighbour (KNN) algorithm, the proposed system utilizes advanced ensemble learning methods, specifically Light Gradient Boosting Machine (Light) and Random Forest. These algorithms are chosen for their capability to handle large-scale data and provide higher detection accuracy. The architecture of the proposed system involves three main stages: feature selection, model training, and intrusion detection. Initially, Correlation-based Feature Selection (CFS) combined with the Bat Algorithm (BA) is applied to reduce dimensional by removing irrelevant or redundant features, thereby improving model efficiency.[19] Following feature selection, the system builds and trains an ensemble classifier using the selected attributes.[17] The classifier then evaluates incoming network data and identifies potential intrusions based on a target variable ("Has Detection") which indicates whether a system has been compromised. This approach ensures a more accurate and saleable intrusion detection system, capable of learning from historical patterns and adapting to complex and evolving network environments.

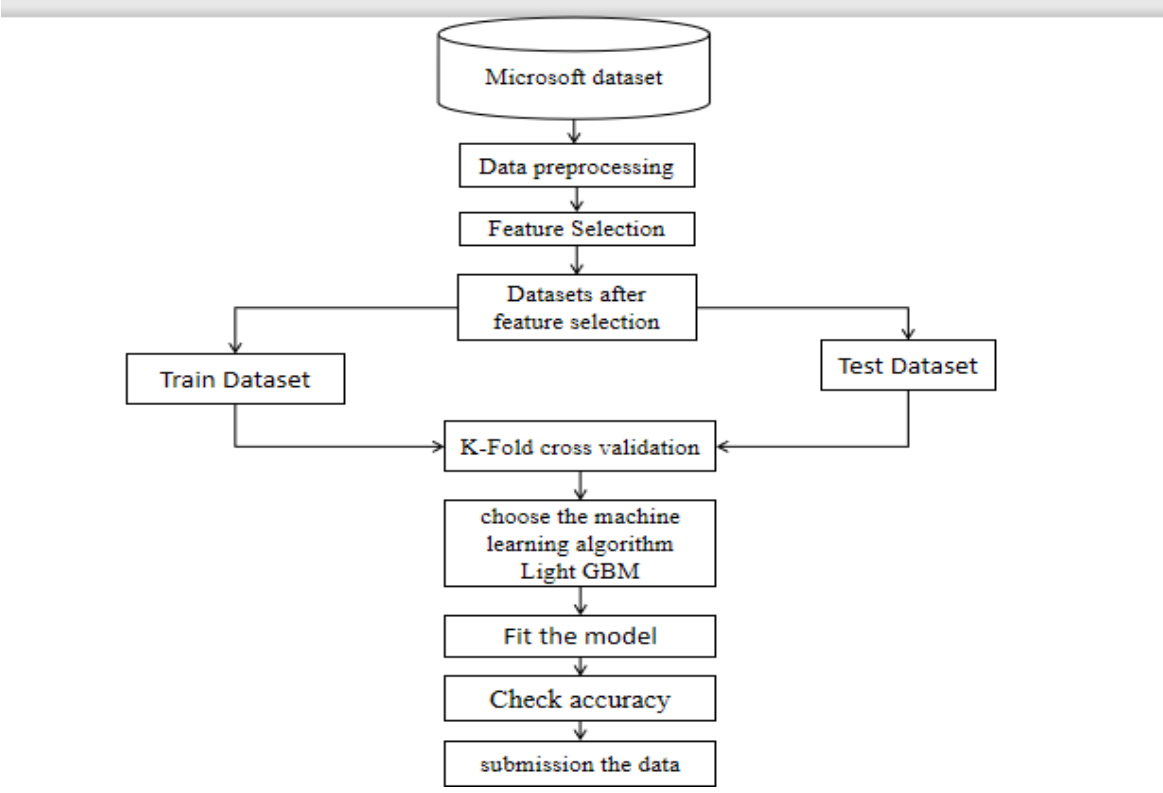

Fig. proposed system

International Scientific Journal of Engineering and Management (ISJEM)
Volume: 04 Issue: 07 | July – 2025
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM04803

UML DIAGRAMS:

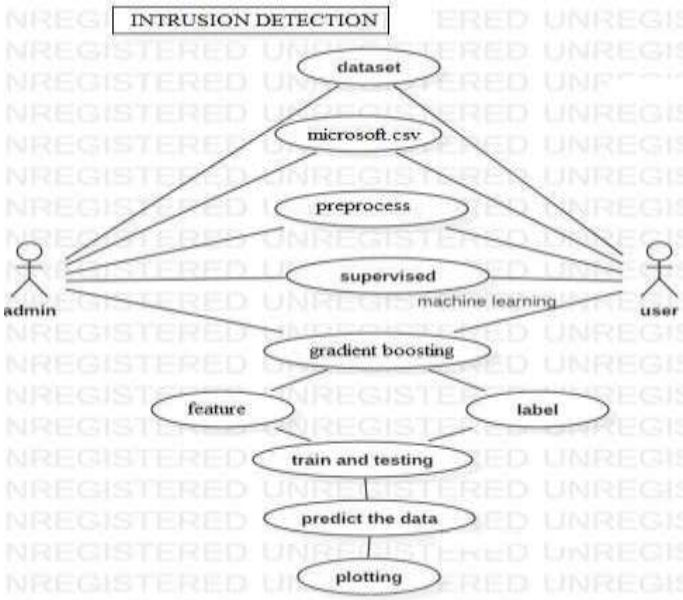

FIG: USE CASE DIAGRAM

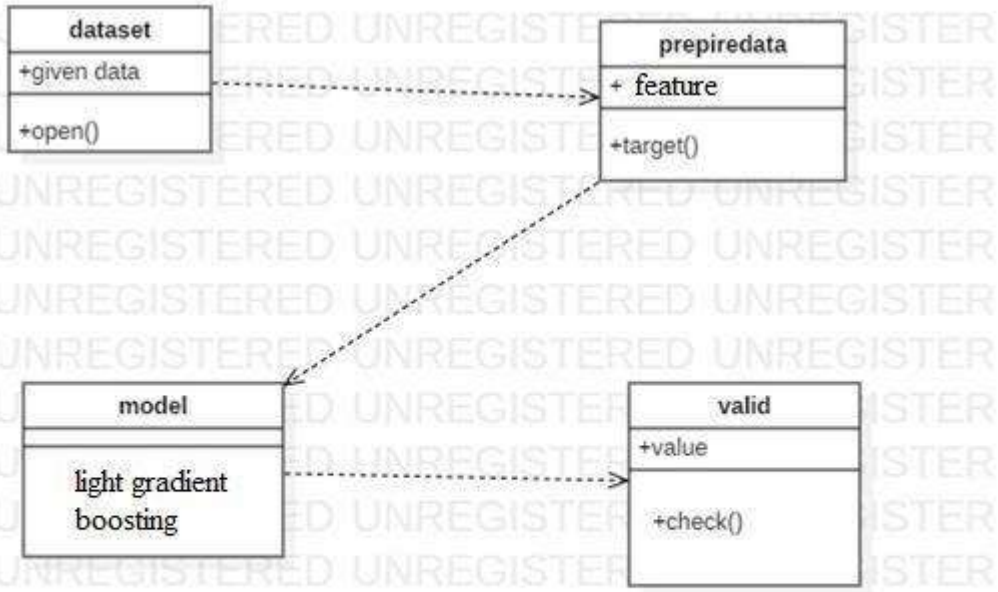

FIG:CLASS DIAGRAM

:

1.1.2 Advantages:

- ✅ **Improved Accuracy:**

Utilizes ensemble learning methods (Light and Random Forest) which combine multiple models to deliver higher intrusion detection accuracy.

- 🚀 **Fast Training and Prediction**:

Light supports faster training and prediction due to its leaf-wise tree growth and optimized performance

ISSN: 2583-6129
DOI: 10.55041/ISJEM04803

- 🧠 **Low Memory Usage**:

Light is optimized to consume less memory, making it suitable for large datasets and resource-constrained environments.

- 📈 **Handles Large-Scale Data**:

Capable of processing big datasets effectively, which is essential for real-world intrusion detection.

- 🔐 **Robust Intrusion Detection**:

Can identify complex and hidden intrusion patterns that may be missed by traditional systems.

Architecture:

The architecture of the proposed system for "Network Intrusion Detection Using Machine Learning" follows a structured multi-stage approach to effectively detect malicious activities in a network.[8] It begins with the collection of telemetry data from Windows machines, where each record consists of several system attributes along with a target label called "Has Detentions," indicating whether malware was detected.[15] This raw data undergoes prepossessing to handle missing values, outliers, and normalization to ensure it is clean and consistent for analysis. Following this, the system applies Correlation-based Feature Selection (CFS) to filter out irrelevant attributes. To further enhance the selection process, the Bat Algorithm (BA) is employed to optimize the feature subset, reducing dimensional and improving model performance. The refined dateset is then used to train ensemble machine learning models, primarily Light and Random Forest. Light is favoured for its high speed, low memory consumption, and scalability.[10] The trained models classify incoming data as either normal or intruded based on the "Has Detection" outcome. Model performance is evaluated using metrics such as confusion matrix, ROC curve, precision, and recall. The final result clearly indicates whether the system has been compromised, enabling timely alerts or actions. This architecture ensures a saleable, accurate, and efficient intrusion detection system that can adapt to evolving network environments.

Algorithm:

The algorithm for the proposed Network Intrusion Detection System using Machine Learning follows a structured 15-step approach. It begins with loading the raw network traffic dataset's, which contains various machine and network-related features. The data is first cleaned by handling missing values and removing anomalies, followed by normalization and encoding of categorical variables.[14] Once the data is pee-processed, feature selection is performed using the Correlation-Based Feature Selection (CFS) method, which identifies the most relevant features that are strongly correlated with the target variable. To further refine the selection, the Bat Algorithm (BA) is applied to optimize the subset of features for better model accuracy. The refined dateset is then split into training and testing sets to evaluate model performance. [11]Two machine learning models—Light and Random Forest—are initialized and trained on the training data. These models are chosen for their efficiency and high performance in classification tasks. Once trained, the models are used to predict intrusion on the test data. The predictions are evaluated using metrics such as accuracy, precision, recall, and ROCOCO to understand the effectiveness of the classifiers.

Techniques:

The techniques used in the project *"Network Intrusion Detection Using Machine Learning"* combine advanced machine learning models, feature selection methods, and ensemble learning strategies to improve the accuracy and reliability of intrusion detection. The primary classification techniques employed are Light Gradient Boosting Machine (Light) and Random Forest, both of which are supervised learning algorithms.[16] Light is a fast, high-performance gradient boosting framework that grows tree leaf-wise and is especially suited for handling large datasets with high dimensional. Random Forest, on the other hand, is an ensemble technique based on decision trees, which reduces over-fitting and enhances generalization by combining the predictions of multiple trees. In addition to classification techniques, the system incorporates feature selection to enhance model performance. The Correlation-Based Feature Selection (CFS) technique is used to select features that are highly correlated with the target variable but uncorrelated with each other, ensuring that only the most informative attributes are retained. To further optimize this selection, the Bat Algorithm (BA)—a meta-heuristic inspired by the echolocation behaviour of bats—is employed.[9] This hybrid approach improves accuracy by reducing noise and dimensional in the dateset.

Tools:

The project "Network Intrusion Detection Using Machine Learning" utilizes a variety of tools to implement and evaluate the intrusion detection system. The primary programming language used is Python, chosen for its simplicity, flexibility, and extensive machine learning libraries. Development and testing were conducted using **Jupiter Notebook**, an interactive environment that supports code execution, visualization, and documentation in one place. For data manipulation and analysis, libraries such as Numpy and Pandas were used to handle arrays and structured datasets efficiently. [18]Visualization tools like Matplotlib and Seaboard were employed to plot graphs such as ROC curves and correlation matrices. Machine learning models were built using Scikit-learn, which provides

ISSN: 2583-6129
DOI: 10.55041/ISJEM04803

implementations of Random Forest, data prepossessing methods, and evaluation metrics. For advanced gradient boosting, Light was integrated to train fast and saleable models. The feature selection process combined **CFS** (implemented via correlation functions) and the Bat Algorithm, which likely required custom Python scripting or optimization libraries. Anaconda served as the overall Python environment manager, offering pre-install packages and a clean workspace. Database interactions (if any) were planned using MySQL connectors or CSV files for structured input. These tools together enabled effective prepossessing, model training, evaluation, and visualization, supporting a full machine learning workflow for detecting network intrusions.

Methods:

The methods used in the *"Network Intrusion Detection Using Machine Learning"* project are centred around a structured machine learning pipeline designed to detect malicious network activities effectively. The process begins with data collection, where telemetry data from machines (like logs and threat reports) are gathered. This raw data is then passed through a data prepossessing stage that includes handling missing values, normalizing features, and encoding categorical data to ensure that it is clean and machine-learning-ready. The next critical step involves feature selection, where a hybrid method combining Correlation-Based Feature Selection (CFS) and the Bat Algorithm (BA) is applied to reduce dimensional and retain only the most informative features. After selecting the optimal features, the dateset is split into training and testing subsets. Two supervised machine learning algorithms—Light and Random Forest—are then trained on the training data. These algorithms are chosen for their high accuracy and ability to handle large, complex datasets efficiently. Once trained, the models are used to predict intrusions on the testing data. To evaluate performance, methods such as the confusion matrix, precision, recall, and ROC curve are applied to understand the strengths and weaknesses of each model.[7] An optional ensemble method like majority voting is used to combine the outputs of both models, further improving prediction reliability. Finally, the trained model is saved and can be used for real-time intrusion detection or further testing.[20] These methods work together in a systematic and layered fashion to achieve accurate and saleable intrusion detection.

METHODOLOGY

Input:

network security by detecting intrusions using machine learning techniques like Limelight and Random Forest. It overcomes the limitations of traditional methods through feature selection (CF S-BA) and provides higher accuracy in identifying malware-infected systems. [5]Built using Python, Jupiter Notebook, and big data tools like Hadoop, the system is user-friendly and thoroughly tested. It concludes that machine learning significantly enhances intrusion detection, with future improvements planned using soft computing for better adaptability.The project *"Network Intrusion Detection Using Machine Learning"* aims to improve

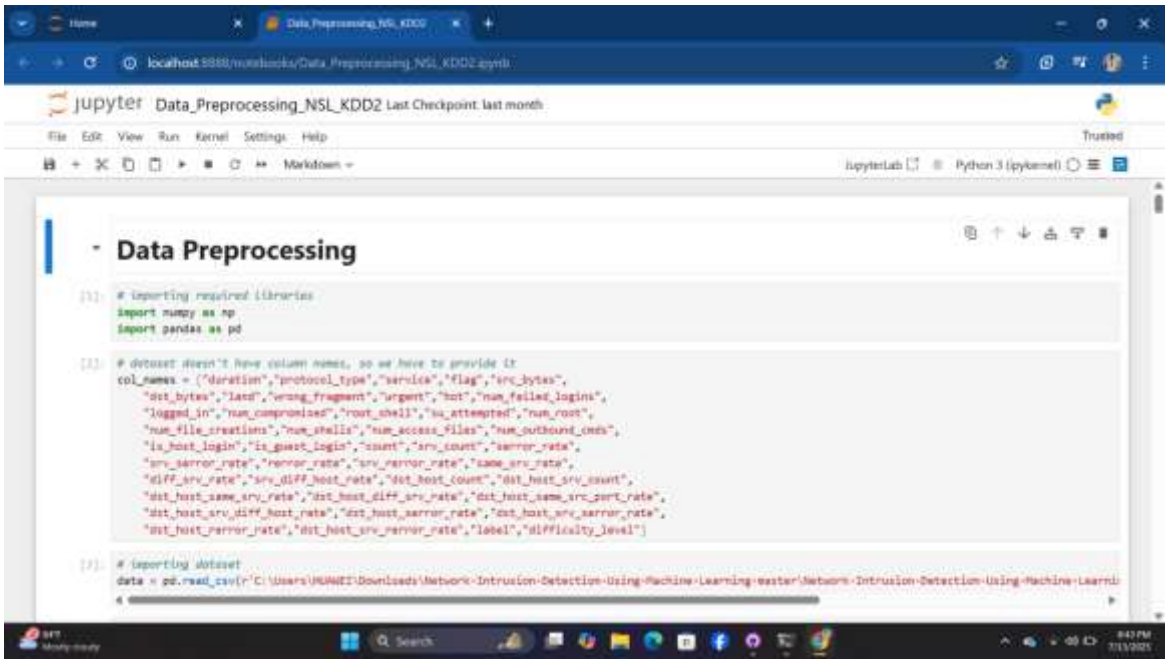

Fig: input data

ISSN: 2583-6129
DOI: 10.55041/ISJEM04803

Method of Process:

The project begins with data collection from Windows Defender telemetry, followed by prepossessing steps such as handling missing values and encoding features. Feature selection is applied using a hybrid method combining Correlation-based Feature Selection (CFS) and the Bat Algorithm (BA) to eliminate irrelevant attributes. The selected data is then used to train machine learning models—primarily Limelight and Random Forest—for intrusion detection.[1] The system is evaluated using confusion matrix, accuracy, and ROC curve metrics. Finally, the model is tested through various testing stages like unit, integration, system, and acceptance testing to ensure reliability and effectiveness in real-world scenarios.

Output:

The output of the project is a machine learning-based intrusion detection system that accurately predicts whether a system is infected or secure. It generates results in binary form—1 for detected intrusions and 0 for no intrusion. The model demonstrates improved accuracy and efficiency through optimized feature selection and ensemble methods like Limelight. Performance is visualized using metrics such as the confusion matrix, ROC curve, and accuracy score, confirming the model's effectiveness in detecting threats within a network environment.

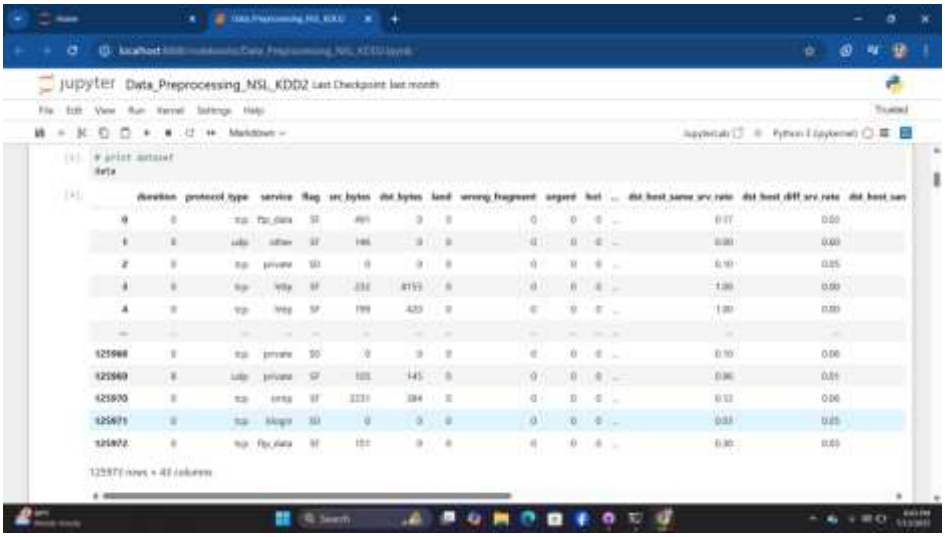

Fig:output data

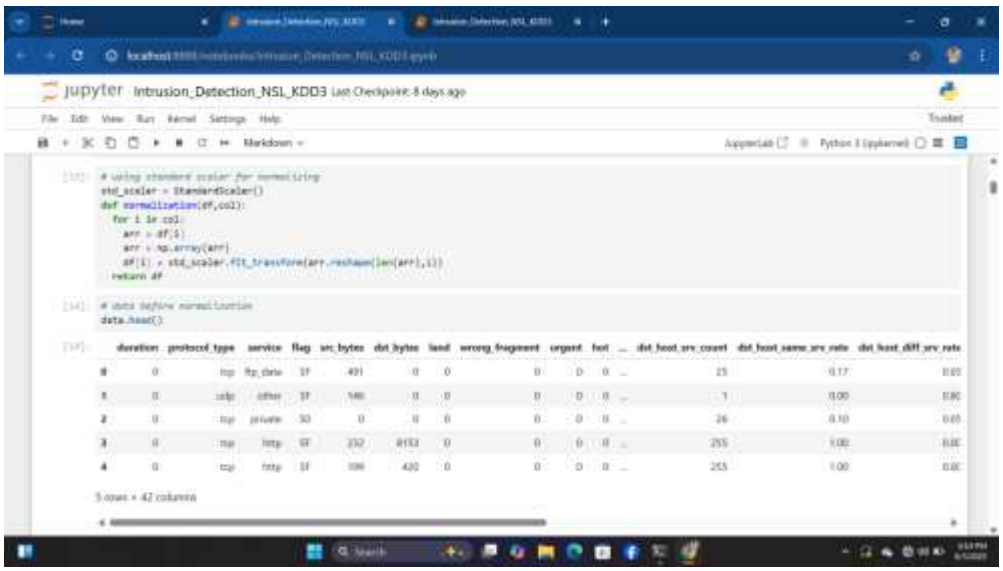

Fig:outputdata

ISSN: 2583-6129
DOI: 10.55041/ISJEM04803

RESULTS:

The results of the project demonstrate that the proposed machine learning-based intrusion detection system significantly improves detection accuracy compared to traditional methods like K-Nearest Neighbor. By using Limelight and Random Forest classifiers along with an optimized feature selection approach (CF S-BA), the system effectively identifies intrusions with improved precision and recall. The Limelight model achieved an accuracy score of approximately 65.6%, showing better performance in handling large-scale data and detecting threats. The use of confusion matrix and ROC curve further validated the model's capability in distinguishing between normal and malicious activity, confirming its practical applicability in real-world network environments.

DISCUSSIONS:

The project explores the use of machine learning techniques, particularly Limelight and Random Forest, to enhance network intrusion detection. Traditional approaches like KNN showed limited performance due to issues with scalability, high-dimensional data, and sensitivity to noise. The integration of advanced feature selection methods such as Correlation-based Feature Selection (CFS) and Bat Algorithm (BA) proved effective in reducing data conditionality and improving model accuracy. The results highlighted that not all features contribute equally to intrusion detection, emphasizing the need for proper feature engineering. The use of real-world telemetry data from Windows Defender ensured the model was tested on realistic scenarios. Overall, the discussion confirms that combining feature selection with ensemble learning models yields more accurate and efficient intrusion detection systems, although there remains room for further optimization and real-time adaptability.

CONCLUSION:

In conclusion, the project successfully demonstrates that machine learning techniques, especially Limelight and Random Forest, can significantly improve the accuracy and efficiency of network intrusion detection systems. By incorporating advanced feature selection methods like CFS and the Bat Algorithm, the system effectively reduces noise and irrelevant data, enhancing detection performance. The model achieves reliable results in predicting whether a system is intruded or not, using real-world telemetry data. This approach not only strengthens cybersecurity defenses but also shows promise for saleable and adaptive deployment. The study confirms that intelligent, data-driven IDS solutions are essential for addressing modern network security challenges.

FUTURE SCOPE:

In the future, the intrusion detection system can be enhanced by implementing soft computing techniques such as genetic algorithms, fuzzy logic, or deep learning models to improve feature selection and classification accuracy. Real-time intrusion detection across dynamic and heterogeneous environments can also be explored, enabling adaptive learning from new types of cyber threats. Additionally, integrating the system with cloud-based monitoring tools and deploying it in large-scale enterprise networks can further validate its scalability and effectiveness. Expanding the dateset and incorporating threat intelligence feeds can improve its robustness against emerging malware and intrusion patterns.

ACKNOWLEDGEMENT:

Chinthagingala Vasundhara: working as an Assistant professor in master of computer application in sanketika vidya parishad engineering college, Visakhapatnam Andhra Pradesh. With 2 years of experience in computer science and engineering (CSE), accredited by NAAC.with her area of intrest in java full stack.She is dedicated and emerging academician in the field of Computer Science, currently serving as a faculty member. With a strong foundation in technical concepts and a passion for teaching, she has begun his academic journey by actively mentoring students in practical and innovative projects. As a faculty, she has already made a significant impact by guiding student teams through their academic projects with clarity, enthusiasm, and technical proficiency. His commitment to student success and interest in research-driven teaching make him a promising contributor to academic excellence

BOBBADI ADARSH is pursuing his final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, zaffiliated by Andhra University and approved by AICTE. With interest in Machine learning Bobbadi Adarsh has taken up him PG project on NETWORK INTRUSION DETECTION SYSTEM and published the paper in connection to the project under the guidance of CH. VASUNDHARA, Assistant Professor, SVPEC.

ISSN: 2583-6129
DOI: 10.55041/ISJEM04803

REFRENCES:

1. https://www.sciencedirect.com/science/article/pii/S0167404822002553
2. https://ieeexplore.ieee.org/abstract/document/8264962/
3. https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4150
4. https://ieeexplore.ieee.org/abstract/document/8406212/
5. https://www.mdpi.com/2076-3417/12/22/11752
6. https://ieeexplore.ieee.org/abstract/document/5504793/
7. https://search.proquest.com/openview/5dad8a026f0873b47b53edb42569bd40/1?pq-origsite=gscholar&cbl=1976342
8. https://www.sciencedirect.com/science/article/pii/S1877050921011078
9. https://www.sciencedirect.com/science/article/pii/S0957417409004801
10. https://arxiv.org/abs/1809.02610
11. https://www.sciencedirect.com/science/article/pii/S0957417421011507
12. https://ieeexplore.ieee.org/abstract/document/8644161/
13. https://pdfs.semanticscholar.org/2f63/79796151a921fc413edde16dd455d7046d21.pdf
14. https://www.sciencedirect.com/science/article/pii/S1084804520302411
15. https://www.sciencedirect.com/science/article/pii/S0045790622000684
16. https://link.springer.com/chapter/10.1007/978-3-319-98842-9_6
17. https://ieeexplore.ieee.org/abstract/document/9491770/
18. https://ieeexplore.ieee.org/abstract/document/8546162/
19. https://ieeexplore.ieee.org/abstract/document/9040718/
20. https://ieeexplore.ieee.org/abstract/document/8268746/

