“INTELLIGENT INTRUSION DETECTION SYSTEM USING MACHINE LEARNING”

SUPERVISED BY

MUTHUKUMARAN .D
Assistant professor, ECE

PRESENTED BY

1. J. MANINDRA (VTU 19769)
2. J. ESWARA VENKATA SAI (VTU 20999)
3. N. YASWANTH SAI (VTU 21029)

ABSTRACT

- With the increasing sophistication of cyber threats, the need for robust and adaptive intrusion detection systems (IDS) has become paramount in safeguarding sensitive digital assets. Traditional rule-based IDS are inadequate in detecting novel and evolving attacks, leading to a demand for intelligent systems capable of learning and adapting to dynamic threats. This paper presents an intelligent intrusion detection system leveraging machine learning (ML) algorithms for enhanced detection accuracy and efficiency. The proposed system integrates various ML techniques, including supervised, unsupervised, and semi-supervised learning, to analyze network traffic patterns and identify anomalous behavior indicative of intrusion attempts. By training on labeled datasets containing both normal and malicious activities, supervised learning algorithms such as Support Vector Machines (SVM), Random Forest, and Neural Networks can effectively classify incoming network traffic in real-time.

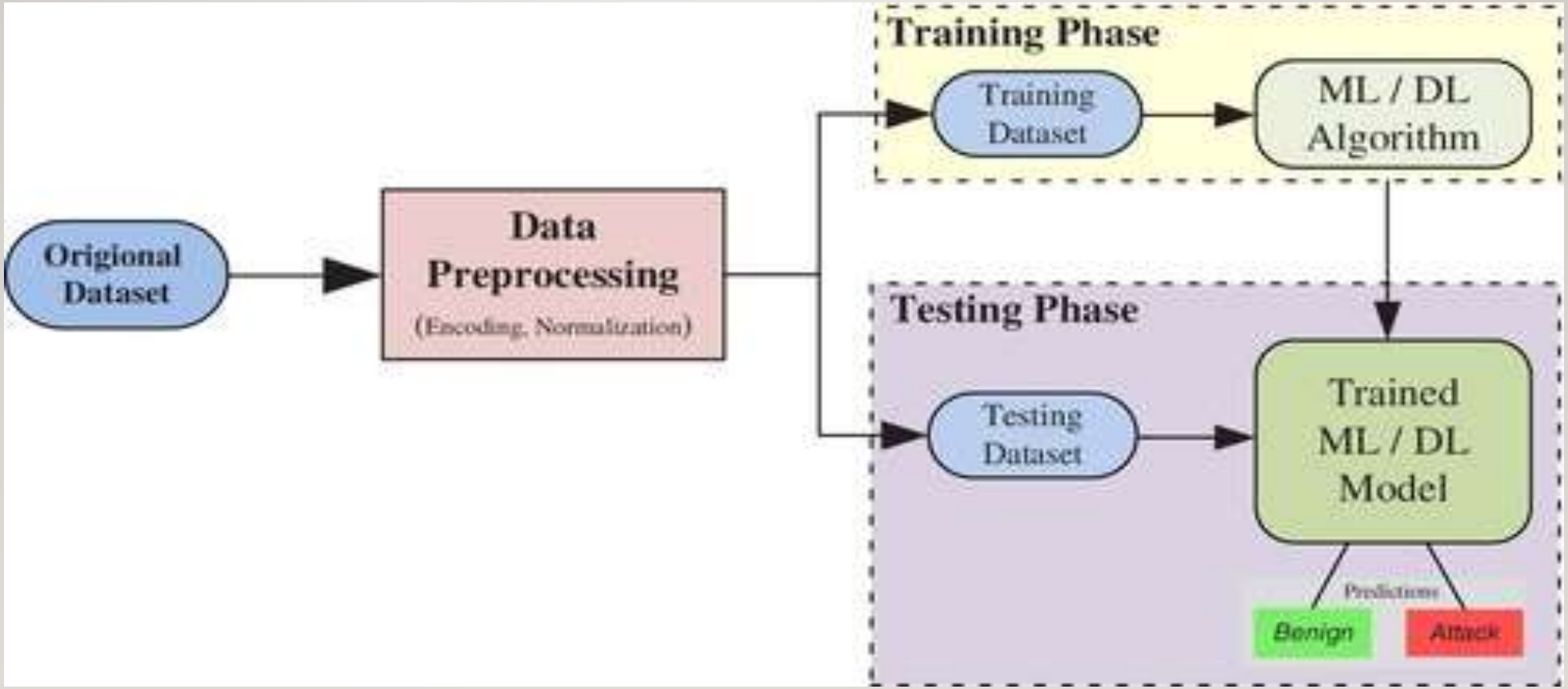
OBJECTIVE

- The objective for an Intelligent Intrusion Detection System using Machine Learning could be to develop a robust and adaptive system capable of accurately identifying and classifying various types of network intrusions in real-time, thereby enhancing the overall security posture of the network.
- **Scope of Project:** Collecting and preprocessing a diverse dataset of network traffic data, including both normal and anomalous behavior.
- **Aim of Project:** Training and fine-tuning machine learning models such as anomaly detection algorithms, supervised classifiers, or deep learning architectures to effectively differentiate between normal and malicious network activity.

INTRODUCTION

- Due to the rapid increase in the number of applications and organizations using computer networks, security is becoming increasingly important. Most companies use network security tools like antivirus and anti-spam software to protect themselves from network attacks. These tools can’t detect complex or new attacks. An IDS enables computer networks and computers to detect and eliminate unwanted intrusions. Identifier systems can collect and process information from various sources within a network or computer, identifying threats that can make people vulnerable, such as misuse and intrusion. IDSs (Intrusion Detection Systems) are systems that continuously monitor and analyze events occurring on a network to detect malicious activity. IDS are now regarded as an important element of the security infrastructure in most companies. By detecting intrusions, companies can deter attacks on their networks. Security professionals could use this method to reduce current network security risks and the complexity of current threats.

BLOCK DIAGRAM



METHODOLOGY

METHODOLOY FOR IDS:

Data Collection and Preprocessing: Gather labeled datasets containing both normal and malicious activities. These datasets should represent the environment in which the IDS will operate.

Preprocess the data: This includes tasks such as data cleaning, normalization, feature extraction, and dimensionality reduction.

Feature Selection: Identify relevant features that can effectively discriminate between normal and malicious activities. This may involve domain knowledge or using techniques like feature importance analysis.

Model Selection: Choose appropriate machine learning algorithms for building the IDS. Common choices include: Supervised learning algorithms (e.g., Decision Trees, Random Forests, Support Vector Machines, Neural Networks) Unsupervised learning algorithms (e.g., K-means clustering, Isolation Forest) Semi-supervised learning algorithms (e.g., Self-training, Co-training) - Consider ensemble methods to combine multiple models for improved performance.

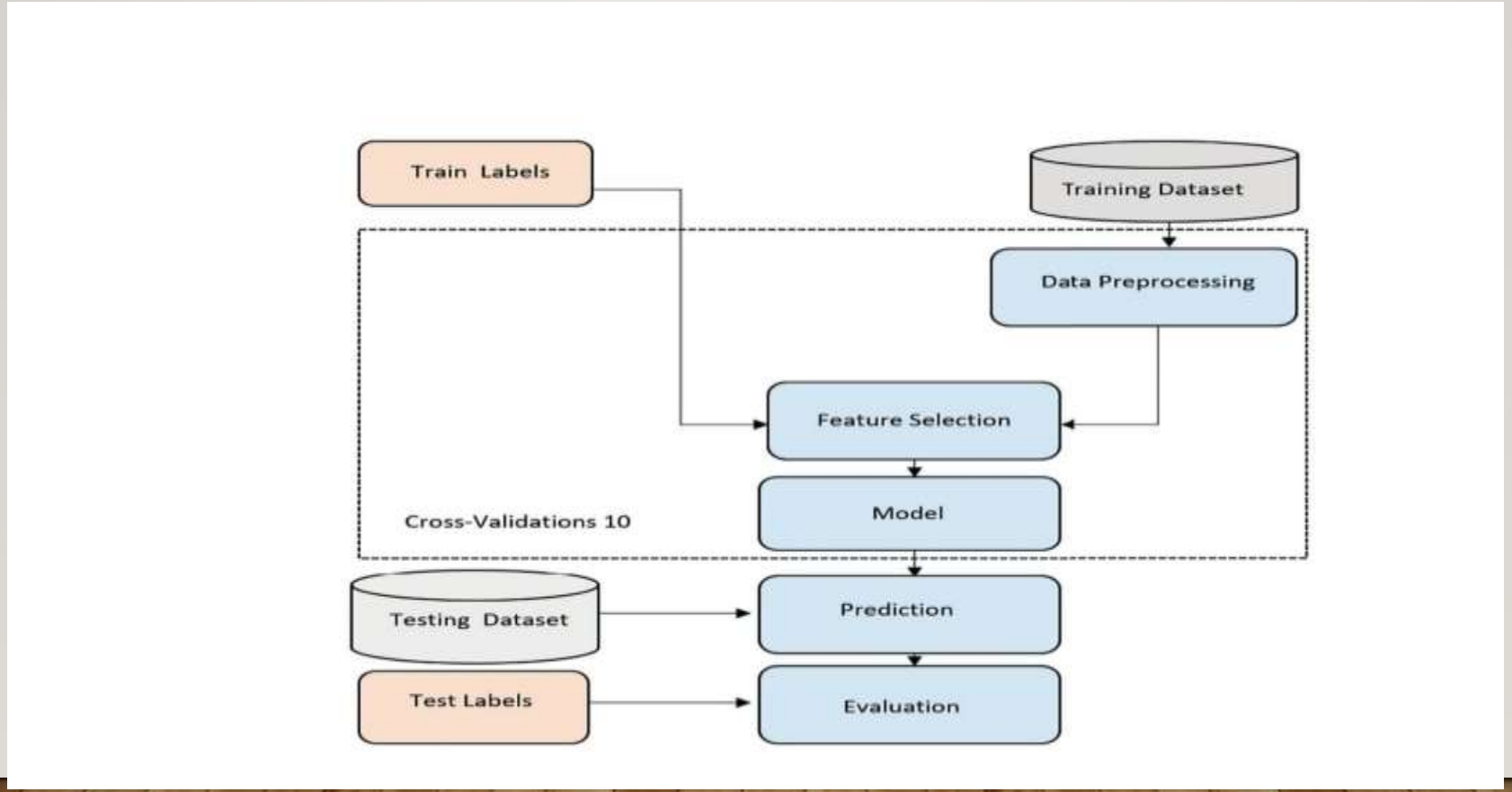
Model Training: Split the labeled dataset into training, validation, and test sets. Train the selected machine learning models using the training data. Tune hyperparameters using the validation set to optimize the model's performance.

Evaluation: Evaluate the trained models using the test set to assess their performance. Calculate metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to measure the effectiveness of the IDS.

Deployment and Monitoring: Integrate the trained model into the IDS infrastructure. Continuously monitor the performance of the IDS in the production environment. Implement mechanisms for model updating and retraining to adapt to evolving threats.

Maintenance and Enhancement: Regularly update the IDS to incorporate new threat intelligence and adapt to emerging attack techniques. Explore advanced techniques such as deep learning, anomaly detection, or adversarial learning to enhance the IDS capabilities.

FLOWCHART


Train Labels
Training Dataset
Data Preprocessing
Feature Selection
Model
Cross-Validations 10
Testing Dataset
Prediction
Test Labels
Evaluation

HARDWARE/SOFTWARE USED

- Software Components:
  - Python Language
  - Google Collab
  - Data Set From Kaggle

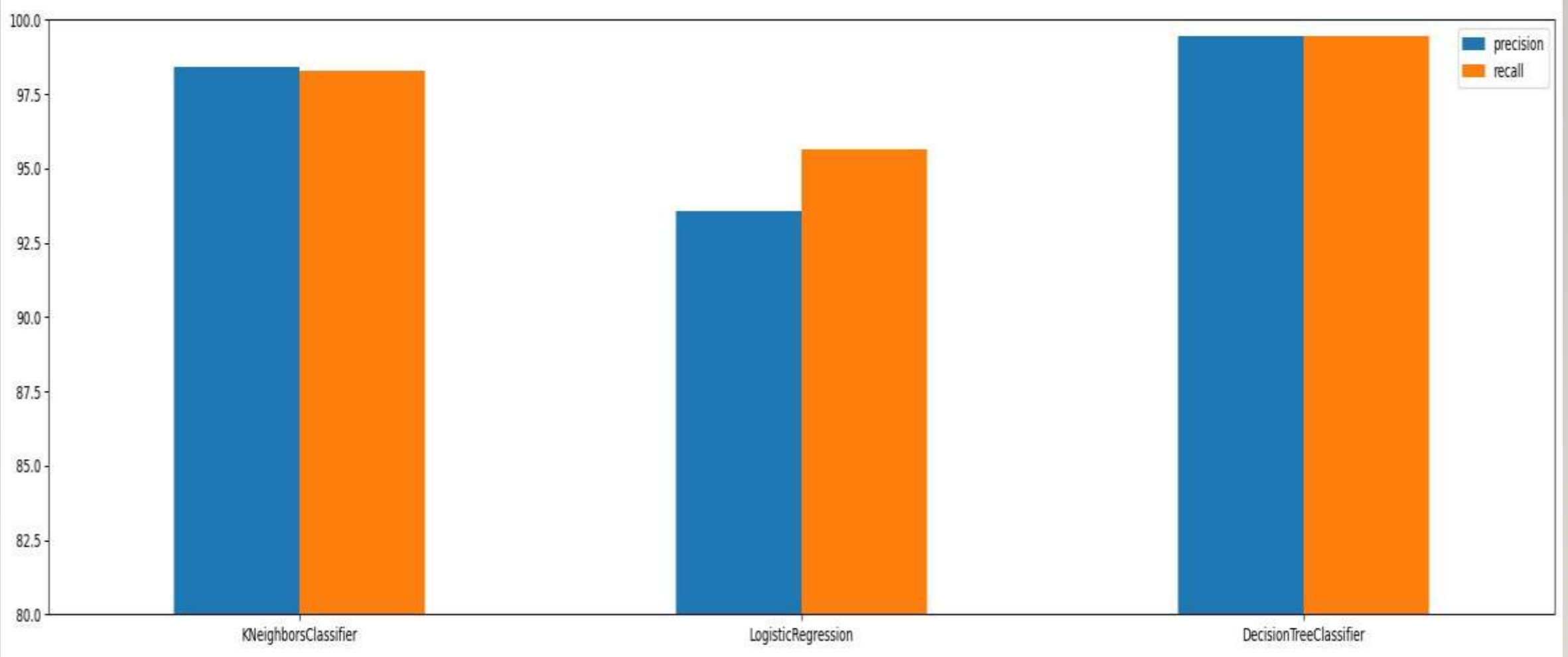
WORKING

• Collect the dataset from the kaggle, dataset must be a KDD-99 or IDS datasets.
• In this we are using the Goggle collab software or anconda navigator in that Jupiter note book software.
• we need to import python libraries like numpy, pandas, seaborn and matplotlib.
• After importing libraries we need to import our machine learning techiques like K-Nearest neighbour, support vector machine, decision tree, then run it.
• Then import the collected datasets in the goggle collab and check for no errors occured, then only we will get the output.
• Train the dataset, test the dataset.
• If the dataset contains string values means, convert it to the int values by one hot encoding method.
• After training the code we need to know the test score and train score.
• In this if we are getting highest accuracy rate for the best of the three techniques then we need to select that machine learning technique.

TIMELINE PLAN

S.No	Project Activity	Description	Date of completion (or) Week number
1.	Literature Survey	Undestanding and collecting the information	Week 1
2	Design	Developing the algorithm for the project	Week 2&3
3	Coding/Implementation	Getting ready with all required tools and algorithms and implementation	Week 3,4 &5
4	Analysis	Evaluate the perform of project	Week 5&6
5	Results and discussion	Checking the result and resolve the errors	Week 6 &7
6	Documentation	Recording the key project details and producing the documentation that are required to implement it successfully	Week 8,9&10
7	Paper publication	Identifying a journal/conference with aims& journal/conference	Week 11&12

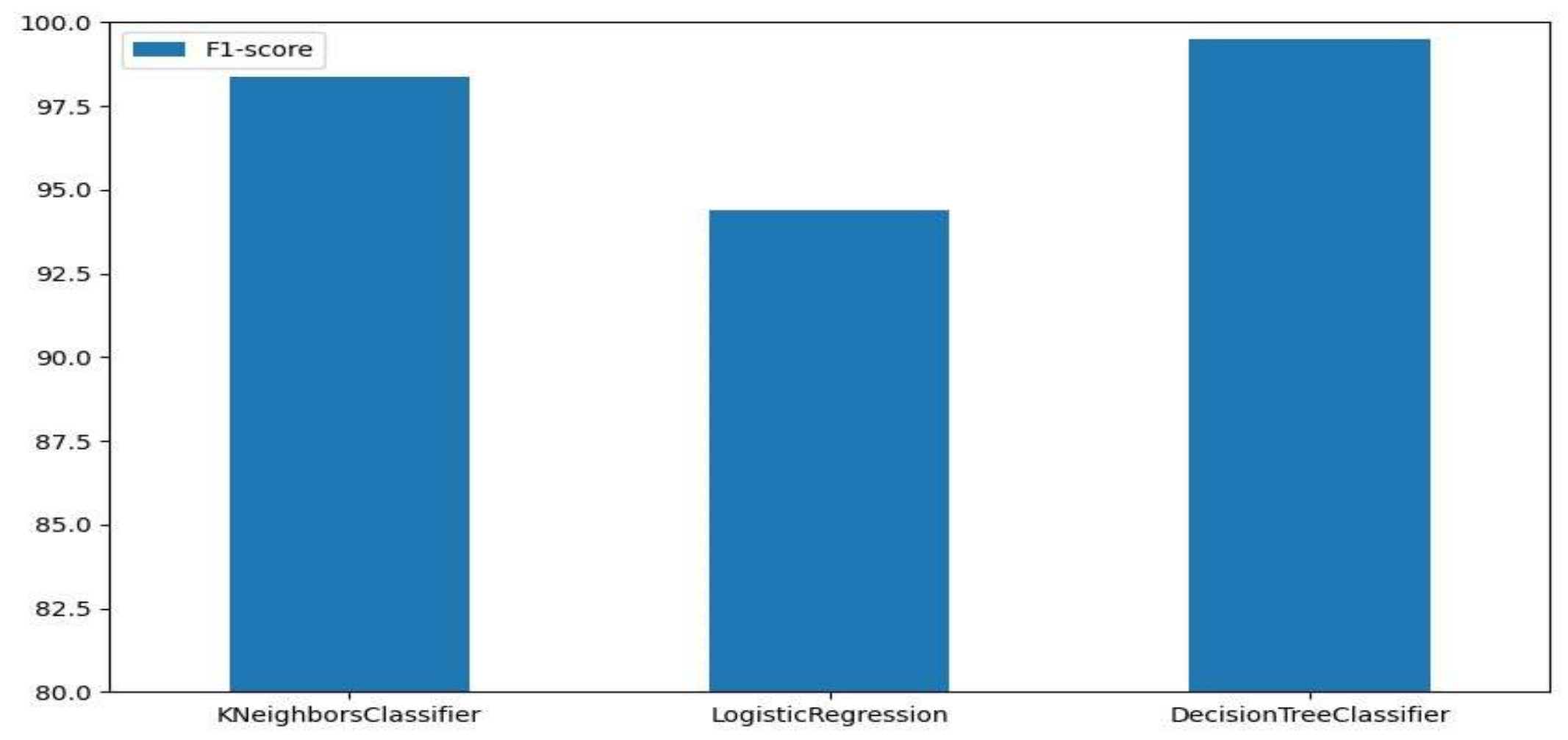
RESULT


100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
precision
recall
KNeighborsClassifier
LogisticRegression
DecisionTreeClassifier

COMPARISION OF MACHINE LEARNING TECHNIQUES

```
*************** KNeighborsClassifier Model Testing ***************
[[3435   63]
 [  65 3995]]
---------------
              precision    recall  f1-score   support

      normal       0.98      0.98      0.98      3498
     anamoly       0.98      0.98      0.98      4060

    accuracy                           0.98      7558
   macro avg       0.98      0.98      0.98      7558
weighted avg       0.98      0.98      0.98      7558

**************** LogisticRegression Model Testing ****************
[[3127  371]
 [ 210 3850]]
---------------
              precision    recall  f1-score   support

      normal       0.94      0.89      0.91      3498
     anamoly       0.91      0.95      0.93      4060

    accuracy                           0.92      7558
   macro avg       0.92      0.92      0.92      7558
weighted avg       0.92      0.92      0.92      7558

************** DecisionTreeClassifier Model Testing **************
[[3484   14]
 [  26 4034]]
-------------
              precision    recall  f1-score   support

      normal       0.99      1.00      0.99      3498
     anamoly       1.00      0.99      1.00      4060

    accuracy                           0.99      7558
   macro avg       0.99      0.99      0.99      7558
weighted avg       0.99      0.99      0.99      7558
```

RESULT


F1-score
100.0
97.5
95.0
92.5
90.0
87.5
85.0
82.5
80.0
KNeighborsClassifier
LogisticRegression
DecisionTreeClassifier

CONCLUSION

The integration of machine learning into Intrusion Detection Systems (IDS) marks a significant evolution in cybersecurity. By harnessing the capabilities of machine learning algorithms, IDS have gained the ability to analyze large volumes of data and detect nuanced patterns indicative of potential security threats. This advancement allows IDS to not only identify known attack signatures but also adapt to emerging threats in real-time, enhancing the overall security posture of organizations. A detailed overview of data mining strategies based on IDS mostly in network is offered in this study. The advantages and disadvantages of these strategies are also examined in order to offer future options for improving intrusion detection performance and thereby improving IDS.

REFERENCES

- Graham, J., Olson, R., & Howard, R. (Eds.). (2011). Cyber security essentials. CRC Press.
- Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16-24.
- Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20), 4396
- Hamid, Y., Sugumaran, M., & Balasaraswathi, V. R. (2016). Ids using machine learning-current state of the art and future directions. Current Journal of Applied Science and Technology, 1-22.

THANK YOU