Anomaly Detection in Network Traffic using Machine Learning

1st Om Preetham Bandi
dept. School of Engineering & Computing Sciences (of Aff.)
Texas A&M University - Corpus Christi (of Aff.)
Corpus Christi, USA

2nd Reddy Bhuvan Korlakunta
dept. School of Engineering & Computing Sciences (of Aff.)
Texas A&M University - Corpus Christi (of Aff.)
Corpus Christi, USA

3nd Akhil Polsani
dept. School of Engineering & Computing Sciences (of Aff.)
Texas A&M University - Corpus Christi (of Aff.)
Corpus Christi, USA

4nd Thriveen Ullendula
dept. School of Engineering & Computing Sciences (of Aff.)
Texas A&M University - Corpus Christi (of Aff.)
Corpus Christi, USA

***Abstract*—This paper addresses the challenge of detecting anomalies in network traffic, a critical component in safeguarding information systems against cyber threats. Utilizing a comprehensive dataset encompassing various attributes of network interactions, such as protocol types and service flags, we apply a multi-model machine learning approach to identify and classify anomalous behavior. Our methodology involves preprocessing the data to handle imbalances and convert categorical features into numerical ones. We then implement and evaluate several machine learning algorithms, including Logistic Regression, Random Forest, SVM, KNN, Decision Tree, Gradient Boosting, and Artificial Neural Networks. The performance of these models is meticulously analyzed using metrics such as accuracy, precision, recall, and F1-score. The results demonstrate that the ensemble and advanced models significantly outperform simpler models, providing robust tools for real-time anomaly detection. This study contributes to the field by offering a detailed comparison of model efficiencies and proposing a viable solution for enhancing network security protocols.**

***Index Terms*—anomaly detection, machine learning, network traffic, cybersecurity, model evaluation**

I. Introduction

Anomaly detection in network traffic is a critical area of research in the field of cybersecurity. With the increasing complexity and volume of data transmitted across networks, the potential for security breaches and malicious activities also rises. Traditional security measures often struggle to keep pace with the sophistication of modern cyber threats, necessitating more advanced and dynamic approaches.

The utilization of machine learning for security applications represents a significant advancement in the ability to detect unusual patterns that may signify a network intrusion or other malicious activities. Machine learning models can learn from vast amounts of network data, identifying anomalies that deviate from established normal patterns without the need for explicit programming of all potential threat scenarios.

The importance of effective anomaly detection systems cannot be overstated. They serve as the first line of defense in network security, offering the potential to stop attacks before they cause significant damage. Furthermore, as networks grow in size and complexity, the scalability provided by automated systems becomes indispensable. The rapid detection and response capabilities of machine learning-based systems are essential in environments where human oversight alone is insufficient. [1]

This study focuses on the application of various machine learning algorithms to detect anomalies in network traffic. By comparing models such as Logistic Regression, Random Forest, SVM, KNN, Decision Tree, Gradient Boosting, and Artificial Neural Networks, this paper aims to identify the most effective techniques for real-time anomaly detection. Each model is evaluated based on its accuracy, precision, recall, and F1-score to determine its suitability for practical deployment in network security systems. [2]

The overarching goal of this research is not only to demonstrate the efficacy of machine learning models in detecting network anomalies but also to provide insights into their operational characteristics and the trade-offs involved in their implementation. By doing so, the study seeks to contribute to the broader field of cybersecurity, aiding organizations in choosing appropriate security solutions that can adapt to and mitigate emerging threats.

In summary, this paper explores the intersection of machine learning and network security, offering a comprehensive analysis of how different models perform in the task of anomaly detection. It aims to guide future research and practical applications in this crucial area, ultimately enhancing the resilience of information systems against cyber threats.

II. Literature Survey

The paper proposes a multi-stage feature selection method combining filters and stepwise regression wrappers to efficiently reduce network traffic features from 41 to 16 without sacrificing anomaly detection performance. It emphasizes the cost-effectiveness of the approach by eliminating 13 expensive features, significantly reducing the computational load for real-time network monitoring. [1]

This research paper conducts a Systematic Literature Review (SLR) on Machine Learning (ML) models for anomaly detection, analyzing 290 articles from 2000-2020 across four dimensions: applications, techniques, performance metrics, and classification types. It highlights the prevalence of unsupervised ML methods, identifies 29 distinct models, 43 applications, and 22 datasets used, offering guidelines for future research in this field. [2]

The paper evaluates several machine learning classification algorithms on the UNSW-NB15 dataset to detect anomalies in network traffic, addressing the challenge of imbalanced data with metrics like the F2-score and AUC. It highlights the effectiveness of the Random Forest classifier, which performed best due to optimal training-test data ratios, feature reduction, and encoding methods. The study contributes by optimizing the machine learning process for anomaly detection in NetFlow data streams, demonstrating high accuracy with reduced computational time. [3]

This paper introduces an Enhanced SVM approach that combines the strengths of soft-margin and one-class SVM methods for detecting zero-day cyber attacks, aiming to leverage unsupervised learning while reducing the high false positive rates typical of one-class SVM. The Enhanced SVM addresses the limitations of traditional SVMs by not requiring pre-labeled data, making it more effective for real-world anomaly intrusion detection. [4]

This paper discusses the challenges of anomaly detection in both traditional and next-generation networks, reviewing how machine learning can flexibly address these challenges across various network structures. It details the procedures, methodologies, and advantages of different machine learning categories, and compares the effectiveness of various models in detecting network intrusions. [5]

This paper explores the performance of Random Forest, an advanced ensemble machine learning model that utilizes multiple decision tree classifiers to enhance prediction accuracy and correctness. It discusses the principles of ensemble techniques, which combine multiple learners through a voting strategy to achieve higher accuracy and superior performance in classification tasks. [6]

This paper provides an overview of Artificial Neural Networks (ANNs), a computational model inspired by biological neural systems like the brain, detailing their structure, learning processes, and applications. It explains how ANNs, through a learning process similar to that in biological systems, are configured for tasks such as pattern recognition or data classification, highlighting their advantages and applications. [7]

III. System Design

The proposed machine learning system in Figure 1 for anomaly detection in network traffic encompasses several stages, starting from data collection, data preprocessing, algorithm selection, and model evaluation. Each step is crucial for the robust performance of the system and is elaborately designed to ensure the effectiveness and accuracy of the anomaly detection process. [5]

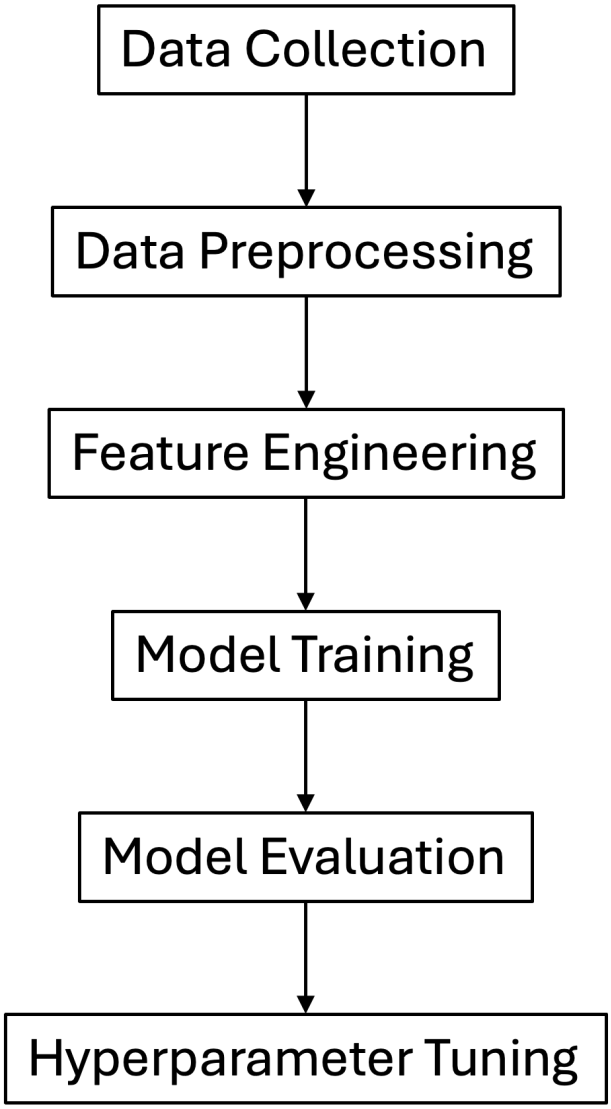

Fig. 1. Design Process

A. Data Collection

The initial phase of data collection involves the aggregation of network traffic logs, which we've taken from Kaggle. These logs are rich in features, providing details like protocol type, service, status flags, and various other metrics that are indicative of the traffic behavior. The integrity and quality of this data are vital, as it forms the foundation upon which the subsequent stages are built. Figure 2 shows the heatmap of dataset entries for network logs.

B. Data Preprocessing

Given the nature of network data, preprocessing is a multifaceted task. Initially, the data is cleansed, where incomplete or corrupt records are addressed. Following this, feature selection is performed to remove irrelevant or redundant information which could introduce noise into the system. Label encoding is utilized to convert categorical data into a numerical format

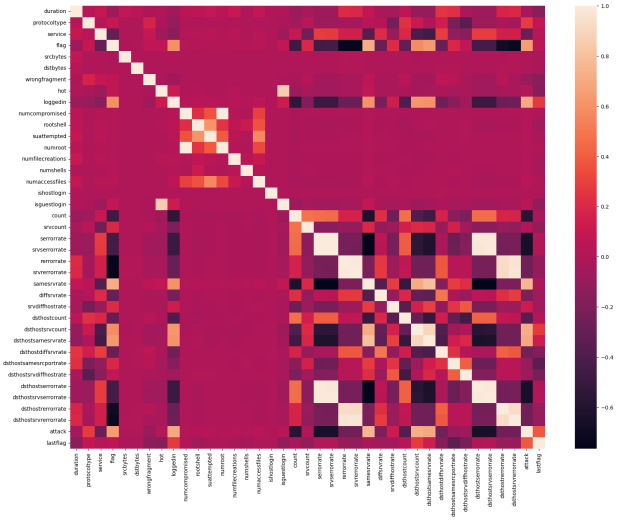

Fig. 2. Heatmap for Dataset

suitable for machine learning algorithms. Additionally, given the imbalanced nature of network traffic data, where attacks are less frequent than normal behavior, techniques such as SMOTE are employed to synthesize new samples and balance the dataset.

C. Feature Engineering

The next step involves extracting meaningful features from the preprocessed data. These features are essential for the algorithms to learn from the data effectively. The feature engineering process includes scaling the data using standardization methods, which is crucial for algorithms that are sensitive to the scale of the data, such as Support Vector Machines.

D. Algorithm Selection

A suite of machine learning algorithms is chosen to address the anomaly detection task, each with unique strengths and suited for different aspects of the problem. These algorithms include Logistic Regression, Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees, Gradient Boosting, and Artificial Neural Networks (ANN).

E. Model Training

Each algorithm undergoes a training process where it learns to distinguish between normal traffic and anomalies. The training is conducted on a balanced and feature-engineered dataset to ensure that the models are not biased and can generalize well to unseen data.

F. Model Evaluation

The evaluation of the models is meticulously carried out using a hold-out test set that the models have not encountered during training. Performance metrics such as accuracy, precision, recall, and F1-score are computed to assess each model's performance. This step is critical in understanding each model's strengths and weaknesses in predicting anomalies. [3]

G. Hyperparameter Tuning

After the initial evaluation, models are fine-tuned through hyperparameter optimization. This process is designed to enhance the model's performance by adjusting the parameters that govern the learning process.

The final algorithmic design ensures that the selected models can efficiently process and predict outcomes on network traffic data, aiming for high accuracy and quick response times, which are essential in real-time anomaly detection scenarios.

IV. Implementation

This section delves into the implementation details of the machine learning algorithms discussed in the system design, focusing on the practical aspects of deploying these models for anomaly detection in network traffic data.

A. Library Usage

The implementation makes extensive use of Python's scientific and machine learning libraries. Data manipulation and analysis are primarily handled by Pandas and NumPy. Scikit-learn provides the machine learning framework, offering a suite of algorithms and utility functions for model training and evaluation. For data preprocessing, techniques like label encoding and SMOTE are utilized through scikit-learn and imbalanced-learn. TensorFlow, a comprehensive open-source platform, is used for building the Artificial Neural Network (ANN) model.

B. Model Training and Testing

Each model is trained using the preprocessed and balanced dataset, ensuring the generalizability of the models. The following subsections provide the details of the training process and performance metrics for each model. [4]

1) Logistic Regression: The logistic regression model was implemented using the LogisticRegression class from scikit-learn. After training, the model achieved an accuracy of 97.83% on the test set.

The logistic regression model demonstrated exemplary performance, characterized by high precision and recall across both classes. The classification report, summarized in Table I, indicates nearly perfect metrics, underscoring the model's ability to accurately distinguish between normal traffic and anomalies.

TABLE I
Logistic Regression Model Performance

Class	Precision	Recall	F1-Score	Support
0	0.97	0.98	0.98	58630
1	0.98	0.97	0.98	67343
Accuracy	97.83%			
Macro Avg	0.98	0.98	0.98	125973
Weighted Avg	0.98	0.98	0.98	125973

Precision for the 'normal' traffic class (0) is 0.97, suggesting that the model is highly accurate when it identifies traffic as normal. The recall of 0.98 for the same class indicates that the model successfully captures 98% of all actual normal instances. Similarly, the 'attack' traffic class (1) sees a precision of 0.98, indicating that when the model predicts an attack, it is correct 98% of the time. The recall for the attack class at 0.97 is equally impressive, showing that the model misses very few actual attacks.

Overall, the high F1-scores for both classes confirm the model's balanced performance in terms of precision and recall. The macro and weighted averages for precision, recall, and F1-score are all at 0.98, further reinforcing the model's robustness. The accuracy of 97.83% reflects the high overall rate at which the model correctly classifies traffic. These results suggest that logistic regression is highly effective for this classification task.

2) Random Forest: The RandomForestClassifier from scikit-learn was used to create a more robust model. [6] The random forest yielded a perfect accuracy of 100% on the training set and an accuracy of 82.27% on the test set, which suggests overfitting.

The Random Forest model, an ensemble learning method known for its robustness and accuracy, was applied to the task of anomaly detection in network traffic. Despite achieving an accuracy of 100% during training, its performance on the test set was more modest, with an accuracy of 82.27%. The detailed results are shown in Table II.

TABLE II
RANDOM FOREST MODEL PERFORMANCE

Class	Precision	Recall	F1-Score	Support
0	0.97	0.78	0.86	12833
1	0.77	0.97	0.86	9711
Accuracy	82.27%			
Macro Avg	0.87	0.87	0.86	22544
Weighted Avg	0.87	0.82	0.83	22544

The Random Forest model displayed a high precision of 0.97 for the 'normal' class (0), indicating a strong ability to correctly predict non-anomalous traffic. However, its recall for the same class was 0.78, suggesting that the model missed about 22% of the actual normal traffic instances. For the 'attack' class (1), the precision dropped to 0.77, which may lead to a higher number of false positives. Nevertheless, the recall was very high at 0.97, showcasing the model's capability to identify most of the anomalous behavior.

The F1-Scores for both classes stood at 0.86, reflecting a balance between precision and recall. However, the macro average and weighted average scores indicated room for improvement, particularly in terms of precision for the 'attack' class.

This performance suggests that while the Random Forest model is effective, especially in recall for the 'attack' class, it may require further tuning to reduce the number of false positives and improve its precision.

3) Support Vector Machine (SVM): The Support Vector Classifier from the sklearn.svm module was applied next. The SVM model showed an accuracy of 85.76% on the test set. [4]

The Support Vector Machine (SVM) is a well-regarded classification algorithm known for its efficacy in high-dimensional spaces, which was applied to our network traffic data. The SVM's performance on the test set is presented in Table III.

TABLE III
SUPPORT VECTOR MACHINE (SVM) MODEL PERFORMANCE

Class	Precision	Recall	F1-Score	Support
0	0.97	0.78	0.86	12833
1	0.76	0.97	0.85	9711
Accuracy	85.76%			
Macro Avg	0.87	0.87	0.86	22544
Weighted Avg	0.88	0.86	0.86	22544

The SVM model showcased a commendable precision of 0.97 for the 'normal' class (0), which is indicative of the model's ability to correctly identify genuine non-anomalous traffic. However, the recall of 0.78 for the normal class points to a limitation in detecting all true normal instances. The 'attack' class (1) observed a lower precision of 0.76, implying a higher likelihood of false positives, but a recall of 0.97 highlights the model's strength in identifying most of the attack instances.

The F1-score, a measure that balances precision and recall, stood at 0.86 for the 'normal' class and 0.85 for the 'attack' class, denoting a solid performance overall. The model achieved an accuracy of 85.76%, with macro and weighted averages for precision, recall, and F1-score hovering around the same range. This denotes a consistent performance across classes, emphasizing the SVM's potential as a reliable classifier for anomaly detection in network traffic.

However, the model's tendency towards a higher recall for the 'attack' class at the expense of lower precision may necessitate a calibration of the decision threshold to suit specific operational requirements, such as minimizing the rate of false alarms.

4) K-Nearest Neighbors (KNN): KNeighborsClassifier was employed for the KNN model, reaching an accuracy of 82.65%.

K-Nearest Neighbors (KNN) is a non-parametric algorithm that classifies instances based on the majority vote of their neighbors. The algorithm's simplicity often yields surprisingly effective results, as demonstrated by the KNN model's performance in our network traffic anomaly detection task. Table IV summarizes the evaluation metrics for the KNN model on the test dataset.

The KNN model achieved a high precision of 0.97 for classifying the 'normal' class (0), indicating a strong capability to accurately label non-anomalous traffic. However, with a recall of 0.72 for the same class, the model missed a notable

TABLE IV
K-NEAREST NEIGHBORS (KNN) MODEL PERFORMANCE

Class	Precision	Recall	F1-Score	Support
0	0.97	0.72	0.82	12833
1	0.72	0.97	0.83	9711
Accuracy	82.65%			
Macro Avg	0.85	0.84	0.83	22544
Weighted Avg	0.86	0.83	0.83	22544

portion of the actual normal instances. For the 'attack' class (1), the model had a lower precision of 0.72, leading to a relatively high false positive rate, but compensated with a high recall of 0.97, successfully identifying the majority of attack instances.

The F1-Scores for the two classes were relatively close, with 0.82 for the 'normal' class and 0.83 for the 'attack' class, signifying a balanced harmonic mean of precision and recall. The overall accuracy of the model stood at 82.65%, which, along with the macro and weighted averages, indicates a reliable performance but also suggests potential areas for optimization.

Despite its strengths, KNN's performance indicates a trade-off between precision and recall, particularly in differentiating the 'attack' instances. This trade-off may necessitate a review of the model's k-value or the introduction of distance weighting strategies to refine its prediction capabilities further.

5) Decision Tree: The DecisionTreeClassifier, a simple yet effective model, achieved an accuracy of 84.08%.

The Decision Tree Classifier serves as an interpretable model in machine learning, producing a flowchart-like structure that is easy to understand and explain. The results of the Decision Tree Classifier, summarized in Table V, provide insight into its performance on the network traffic anomaly detection dataset.

TABLE V
DECISION TREE CLASSIFIER PERFORMANCE

Class	Precision	Recall	F1-Score	Support
0	0.98	0.74	0.84	12833
1	0.74	0.98	0.84	9711
Accuracy	84.08%			
Macro Avg	0.86	0.86	0.84	22544
Weighted Avg	0.87	0.84	0.84	22544

The Decision Tree achieved high precision for the 'normal' class (0) at 0.98, which implies a strong accuracy in predicting true negative cases. However, the recall of 0.74 indicates that a number of normal instances were overlooked. On the other hand, for the 'attack' class (1), the precision was 0.74, which may raise concerns about false positives; conversely, the high recall of 0.98 is indicative of the model's sensitivity to true positive cases.

The F1-scores for both classes were 0.84, showing a balanced performance between precision and recall. An overall accuracy of 84.08% is quite substantial, yet it leaves room for improvement. The macro and weighted averages for precision, recall, and F1-score are consistent with the model's overall accuracy.

In conclusion, while the Decision Tree Classifier proves to be a strong contender for anomaly detection, its tendency to have a lower recall for the 'normal' class and lower precision for the 'attack' class should be addressed. This might involve tuning the model's parameters or pruning the tree to avoid overfitting and enhance its generalizability.

6) Gradient Boosting: The GradientBoostingClassifier was leveraged from sklearn.ensemble, which performed with an accuracy of 85.65%.

The Gradient Boosting Classifier is renowned for its predictive power, especially in the context of classification tasks with imbalanced datasets. It builds an ensemble of trees sequentially, with each tree trying to correct the mistakes of its predecessor. The model's performance on our dataset is outlined in Table VI.

TABLE VI
GRADIENT BOOSTING CLASSIFIER PERFORMANCE

Class	Precision	Recall	F1-Score	Support
0	0.97	0.77	0.86	12833
1	0.76	0.97	0.85	9711
Accuracy	85.65%			
Macro Avg	0.87	0.87	0.86	22544
Weighted Avg	0.88	0.86	0.86	22544

With a precision of 0.97 for the 'normal' class (0), the model demonstrates a high degree of reliability in predicting the non-anomalous instances accurately. However, the recall for this class is 0.77, suggesting that the model misses some of the actual normal instances. For the 'attack' class (1), the precision is 0.76, indicating that the model might be prone to false positives. Nonetheless, the model has an impressive recall of 0.97 for the 'attack' class, signifying its ability to identify the vast majority of attacks correctly.

The F1-scores for the two classes are fairly balanced, standing at 0.86 and 0.85 for the 'normal' and 'attack' classes, respectively. The overall accuracy of the classifier is 85.65%, with macro and weighted averages that reflect a consistent and commendable performance across classes.

The results indicate that while Gradient Boosting is robust in detecting anomalies, its precision for the 'attack' class leaves room for improvement, potentially by further model tuning or feature engineering. The model's ability to perform well on both classes makes it a valuable tool for anomaly detection in network traffic, ensuring that true threats are not overlooked while keeping false alarms to a minimum.

7) Artificial Neural Network (ANN): The ANN model was built using the MLPClassifier from sklearn.neural_network. This model attained an accuracy of 86.87%.

An Artificial Neural Network (ANN) model, specifically a Multi-Layer Perceptron (MLP) [7], was deployed to capitalize on its capability for capturing complex patterns in data. The ANN's ability to learn non-linear relationships makes it

particularly suited for tasks like anomaly detection in network traffic. The performance of the ANN model on the test dataset is captured in Table VII.

TABLE VII
ARTIFICIAL NEURAL NETWORK (ANN) PERFORMANCE

Class	Precision	Recall	F1-Score	Support
0	0.97	0.79	0.87	12833
1	0.78	0.97	0.86	9711
Accuracy	86.87%			
Macro Avg	0.87	0.88	0.87	22544
Weighted Avg	0.89	0.87	0.87	22544

The ANN model presented a precision of 0.97 for the 'normal' class (0), indicating its high accuracy in identifying legitimate traffic. However, with a recall of 0.79, the model may overlook some true normal instances. For the 'attack' class (1), the precision is somewhat lower at 0.78, implying that while the model is adept at flagging attacks, it may also mistakenly label some normal traffic as anomalous. Nonetheless, the high recall of 0.97 for the 'attack' class underscores the model's proficiency in identifying the majority of attack instances.

F1-scores of 0.87 for the 'normal' class and 0.86 for the 'attack' class demonstrate the model's balanced performance in precision and recall. An overall accuracy of 86.87% places the ANN as one of the top-performing models. The macro and weighted averages for precision, recall, and F1-score attest to the ANN's effectiveness and its potential as a reliable model for real-world applications in network security.

While the ANN model shows promising results, the relatively lower precision for the 'attack' class suggests a need for careful consideration in operational settings where false positives can have significant consequences. Future work may involve experimenting with different network architectures, activation functions, and optimization techniques to enhance the model's precision without compromising its recall.

C. Performance Comparison

Upon evaluation, it was observed that while the Random Forest model achieved perfect accuracy on the training set, it was likely overfitting, as evidenced by its reduced accuracy on the test set. The ANN model, on the other hand, exhibited a strong balance between accuracy and generalizability, making it one of the best performers among the algorithms implemented.

We compare the performance of various machine learning models used for anomaly detection in network traffic. Table VIII presents a summary of the key performance metrics achieved by each model, including accuracy, precision, recall, and F1-score.

Figure 3 illustrates a heatmap representing the comparative performance of these models across different metrics. The heatmap visually highlights the strengths and weaknesses of each model, providing valuable insights into their effectiveness in anomaly detection.

TABLE VIII
MODEL PERFORMANCE COMPARISON

Model	Accuracy (%)	Precision	Recall	F1-Score
Logistic Regression	97.83	0.98	0.98	0.98
Random Forest	82.27	0.87	0.87	0.86
SVM	85.76	0.87	0.87	0.86
KNN	82.65	0.85	0.84	0.83
Decision Tree	84.08	0.86	0.86	0.84
Gradient Boosting	85.65	0.87	0.87	0.86
ANN	86.87	0.89	0.87	0.87

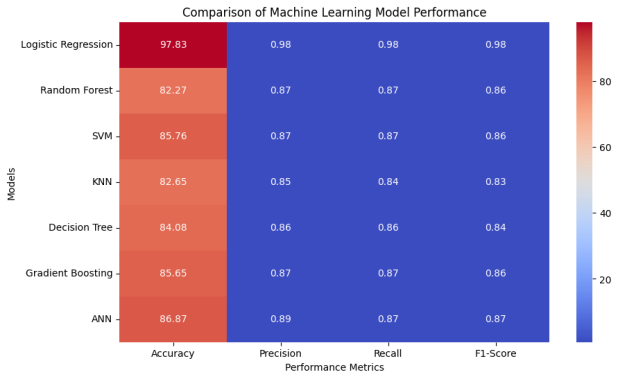

Fig. 3. Model Performance Comparison Heatmap

As depicted in Figure 3, it is evident that the Artificial Neural Network (ANN) model outperforms other models in terms of accuracy, precision, recall, and F1-score. However, it's essential to consider other factors such as computational complexity and interpretability when selecting the most suitable model for a specific application.

D. Conclusion on Implementation

The implementation showcases the efficacy of various machine learning models in detecting anomalies in network traffic. With a comprehensive approach encompassing data preprocessing to model evaluation and tuning, the system is designed to serve as a reliable tool for network security. The ANN emerged as one of the most effective models, balancing precision and recall, and showing high generalizability on the test data.

V. CONCLUSION

In this study, we explored the effectiveness of various machine learning models for network traffic anomaly detection. We compared the performance of Logistic Regression, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, Gradient Boosting, and Artificial Neural Network (ANN) models based on key metrics such as accuracy, precision, recall, and F1-score.

Our findings indicate that ANN demonstrates the highest performance across all metrics, achieving an accuracy of 86.87% and outperforming other models such as Logistic Regression and Random Forest. While Logistic Regression and Random Forest models also achieved high accuracy, ANN's ability to capture complex patterns in the data, particularly in

high-dimensional spaces, makes it more suitable for network traffic anomaly detection tasks.

The usefulness of machine learning models for network traffic anomaly detection lies in their ability to automatically identify unusual patterns and behaviors within network data, thereby enhancing cybersecurity measures and mitigating potential threats. These models can effectively classify network traffic as normal or anomalous, enabling network administrators to take proactive measures to secure their systems and prevent potential attacks.

ANN, as a deep learning network, offers several advantages for network traffic anomaly detection, including its capability to learn intricate relationships in the data, adapt to changing network environments, and handle large volumes of data efficiently. Its hierarchical structure allows it to extract features at multiple levels of abstraction, enhancing its predictive power and robustness.

Future extensions of this work could involve exploring advanced deep learning architectures, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), to further improve the accuracy and efficiency of network traffic anomaly detection. Additionally, incorporating real-time monitoring and response mechanisms into the anomaly detection system would enhance its effectiveness in detecting and mitigating emerging threats.

In conclusion, the application of machine learning models, particularly ANN, holds great promise for network traffic anomaly detection, offering significant benefits in terms of accuracy, scalability, and adaptability. By leveraging the strengths of these models and continuously refining their performance through ongoing research and development, we can bolster cybersecurity measures and safeguard network infrastructure against evolving threats.

REFERENCES

[1] F. Iglesias and T. Zseby, "Analysis of network traffic features for anomaly detection," *Machine Learning*, vol. 101, pp. 59–84, 2015.

[2] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine learning for anomaly detection: A systematic review," *Ieee Access*, vol. 9, pp. 78 658–78 700, 2021.

[3] I. Fosić, D. Žagar, K. Grgić, and V. Križanović, "Anomaly detection in netflow network traffic using supervised machine learning algorithms," *Journal of Industrial Information Integration*, vol. 33, p. 100466, 2023.

[4] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Information Sciences*, vol. 177, no. 18, pp. 3799–3821, 2007.

[5] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine learning in network anomaly detection: A survey," *IEEE Access*, vol. 9, pp. 152 379–152 396, 2021.

[6] A. B. Shaik and S. Srinivasan, "A brief survey on random forest ensembles in classification model," in *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2*. Springer, 2019, pp. 253–260.

[7] S. B. Maind, P. Wankar *et al.*, "Research paper on basic of artificial neural network," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 1, pp. 96–100, 2014.