Detecting malicious IoT network traffic

Refah Tasnia, Sadik Ittesaf Abir, Shah Nadim Kamran Rian, Moriom Islam Mou *

Department of Electrical and Computer Engineering, North South University, Dhaka 1229, Bangladesh

Abstract

In an era defined by the proliferation of Internet of Things (IoT) devices and their seamless integration into our daily lives, data security has emerged as a vital concern. Traditional encryption methods face escalating numbers of threats from evolving cyberattacks, which can affect our daily lives due to this deep interconnection to IoT devices. This study aims to create a system that not only detects malicious IoT network traffic but also analyzes various machine learning algorithms' abilities to identify the network intrusions. The effort ultimately strengthens the security of IoT networks, protects user data, and guarantees device dependability across a variety of applications and sectors. We have used the UNSW-NB15.csv dataset and implemented a diverse set of machine learning algorithms: decision tree, random forest, linear regression, logistic regression, and k-nearest neighbors (KNN). These learning models were applied to predict the two classes normal and abnormal intrusion. We have preprocessed the data, performed feature selection method to improve the predictability and split it into training and testing sets. Each algorithm was trained on the training set and evaluated on the testing set. Evaluation metrics such as accuracy, precision, recall, and F1-score were used to assess the performance of the models. Accuracies of 98.64%, 98.30%, and 97.80% were obtained using Random Forest, KNN and Linear Regression, respectively. Random Forest accomplished the lowest false negatives and highest F Measure with its optimized hyperparameter. This comprehensive approach enabled a detailed comparative analysis, providing valuable insights into the effectiveness of these algorithms for IoT intrusion detection. The unique aspect of this study is the thorough assessment of a variety of models in practical IoT intrusion situations. By conducting extensive experiments, we determine the advantages and drawbacks of each algorithm, revealing detailed insights into when and where particular models perform exceptionally well.

Keywords: IoT (Internet of Things), Machine learning, Network traffic classification, Intrusion detection system (IDS)

1. Introduction

In our rapidly changing world of technology, the Internet of Things (IoT) has become a rising star with the remarkable ability to make our everyday lives "smarter". Serving its aim of accumulating, processing, and transmitting data to the required application, IoT has gained mass popularity and enables physical objects to transfer data via the Internet with zero human interaction [1]. The science behind this wonder is IoT comprising of the sophisticated actuators and chips embedded in those physical objects. However, this rapidly increasing connection between our everyday lives and IoT is also introducing threats to the confidentiality and security of data related to our personal lives, and these dangers can scale up to the organizational level where businesses and different sectors of society are affected [2].

According to a report of IOT World Today, Kaspersky reported 1.51 billion IoT device cyberattacks between January and June 2021, and more than 872 million of them were made aiming at cryptocurrency mining causing distributed denial-of-service (DDoS) shutdowns and breaches of confidential data [3]. Healthcare IoT is an essential part of today's technology which includes wearable health devices, telemedicine, medical equipment, etc. but imagining cyberattacks in such a sensitive and important area of our lives is no less than a terrifying scenario. As per Check Point Research, healthcare organizations all over the globe faced an average of 1463 cyberattacks every week in 2022, an increase of 74% from that in 2021. According to Cynerio's State of Healthcare IoT Device Security 2022 report, 53% of connected healthcare devices are vulnerable to cyberattacks where IV pumps (contributing to 38% of a hospital's IoT footprint) and VoIP systems are at the greatest risk [4]. It is noteworthy that within just 5 minutes of connecting to the internet, IoT devices are vulnerable to attacks while generally, 98% of IoT traffic is not encrypted [5]. IoT networks are highly prized targets of hackers as they act as sources of large ransomware payouts (2 to 4 times higher ransom than other targets) in industries that depend highly on uptime for their survival and in the U.S, it has been found that 61% of all network infringing attempts and 23% of all ransomware attacks were made on Operational Technology (OT) systems [6]. Therefore, our project aims to increase the security of IoT networks which has become an increasingly popular but jeopardized sector of Information Technology, as well as our everyday lives.

Our project uses machine learning techniques to assess IoT network flow parameters and predict whether the network flow is normal or malicious and various ML models have helped the process. Our novelty lies in the meticulous evaluation of these diverse models under real-world IoT intrusion scenarios. Through extensive experimentation, we identify the strengths and limitations of each algorithm, uncovering nuanced patterns and contexts where specific models excel. This nuanced understanding provides valuable insights for practitioners, guiding the selection of optimal intrusion detection techniques in IoT environments, thus enhancing the overall cybersecurity landscape.

Major contributions of this paper include the following.

- Comprehensive Algorithmic Evaluation:
  Our paper offers a thorough comparative analysis of machine learning algorithms, including decision tree, random forest, linear regression, logistic regression, and k-nearest neighbors (KNN) for IoT network intrusion detection.
- Real-World IoT Intrusion Scenarios:
  Our project introduces a realistic and practical dimension by evaluating these algorithms under real-world IoT intrusion scenarios, making the research highly applicable to cybersecurity challenges.
- Strengths and Limitations Unveiled:
  Our research goes beyond high-level comparisons, delving into the nuanced contexts and scenarios where each algorithm excels or encounters limitations, providing practical insights for practitioners.
- Dataset Choice:
  The use of the UNSW-NB15.csv dataset underscores the commitment to realism and practicality in IoT intrusion detection research, reflecting real-world scenarios and challenges.
- Practical Guidance:
  Our findings guide practitioners in selecting optimal intrusion detection techniques in IoT environments, enhancing their ability to address cybersecurity threats effectively. Enhancing Cybersecurity Landscape: Ultimately, Our research contributes to the broader field of cybersecurity, strengthening the overall IoT security landscape by providing valuable, practical insights.

This paper is structured in the following order. Section 2 describes the similar works. The methodology of this project is detailed in Section 3. Section 4 entails the project outcomes and has the visual representations of results. Section 5 discusses project results thoroughly and analyses them. Lastly, Section 6 concludes the paper and discusses possible future works on the project.

2. Related Works

2.1. Related work on IoT Anomaly Detection:

1. Anomaly Detection and Monitoring in IoT Communication: Anomaly detection and monitoring in Internet of Things communication The Internet of Things (IoT) brings new hurdles in identifying anomalies and monitoring all network-connected devices. Furthermore, one of the goals of anonymity in communication is to secure device data traffic. Deris Stiawan and Yazid Idris utilized a heterogeneous device-supporting multi-platform anomaly detection and monitoring system [7]. The suggested solution tackles two issues: (i) how to keep an eye on the network to avoid device malfunctions, and (ii) how to create a feature-rich system for the early identification of anomalies in Internet of Things communication. Mohamed Faisal Elrawy and Ali Ismail Awad presents an in-depth examination of the most recent IDSs created for the IoT paradigm, with an emphasis on the relevant methodologies, features, and procedures [8]. Several publications were investigated in this survey. These publications primarily investigate the design and implementation of intrusion detection systems (IDSs) for usage in the IoT paradigm, which may be implemented in smart environments. The characteristics of all IDS approaches reported in these publications have been summarized. This article also delves into the IoT architecture, new security vulnerabilities, and their relationship to the IoT architecture's layers. This paper made some recommendations for designing an IDS for the IoT, such as the need for a powerful and lightweight system with an appropriate placement strategy that does not jeopardize the integrity, confidentiality, and availability of the IoT environment. This study revealed the necessity for an integrated IDS that can be used in IoT-based smart settings.

2.2. Related work on AI-Driven IoT Security

Abebe Diro and Naveen Chilamkurti used jointly developing Machine Learning-Based models for Anomaly Detection in IoT [9]. They presented an in-depth overview of previous work in building anomaly detection systems for securing an IoT system using machine learning. They also show that blockchain-based anomaly detection systems may develop successful machine learning models jointly. Because of their resource capabilities and in-perimeter position, the applications of anomaly detection systems utilizing machine learning in I.T. systems outperformed the IoT ecosystem. Nonetheless, the present machine learning-based anomaly detection is subject to adversarial assaults. The importance of anomaly detection, the obstacles of building anomaly detection systems, and an analysis of the machine learning methods utilized are all discussed. Bambang Susilo and Riri Fitri Sari explored several machine-learning and deep-learning methodologies, as well as standard datasets, for increasing IoT security performance [10]. Using a deep-learning algorithm, they created an algorithm for identifying denial-of-service (DoS) assaults. They discovered that a deep-learning model might improve accuracy, allowing for the most effective mitigation of threats on an IoT network. They found Random forests and the CNN provided the best result in terms of accuracy and the AUC for multiclass classification.

Shahid Allah Bakhsh and Fawad Ahmed showed an approach to improving IoT network security via IDS by applying deep learning algorithms [11]. This research proposed three models: Feed Forward Neural Networks (FFNN), Long Short-Term Memory (LSTM), and Random Neural Networks (RandNN). They offer exclusive advantages. Attaining a higher degree of precision in identifying intrusions in IoT networks can enhance their security. Harikeish Fowdur and Sandhya Armoogum used subsampling and Machine learning model such as Random Forest to enhance IoT malware detection [12]. This research gives focus on enhancing IoT malware detection. This approach uses data preprocessing using sub sampling and various machine learning models such as Random Forest, Naive Bayes and SVM which provides valuable insights into the detection of malicious IoT traffic. This paper also exposes the limitations related to generalization, model interpretability, and the dynamic nature of cybersecurity threats. R. D. Pubudu L. Indrasiri1 and Ernesto Lee suggested that in order to enhance the detection of malicious traffic in multi domain datasets, the EBF model is effective [13]. Statistical tests and cross validation is confirmed the statistical significance of EBF's performance.

Table 1
Comparison of similar works

Papers	Ref.	Approach	Advantages	Limitations
Anomaly Detection and Monitoring in IoT Communication	[7]	Heterogeneous Anomaly Detection and Monitoring System	Addresses network monitoring and anomaly identification in IoT communication	It has lacking of real-time threat detection capability and adaptability to diverse IoT devices
Machine Learning-Based Anomaly Detection in IoT	[9]	Developing ML models for anomaly detection for IoT system security based on blockchain	Comprehensive analysis of ML-based anomaly detection in IoT emphasizing on the relevant methodologies, features, and procedures	Does not experiment on a wide range of applications and environments
Intrusion Detection Systems (IDS) for IoT-based Smart Environments	[8]	Survey on IoT-Based IDS for Smart Environments	Highlights the ongoing challenge of creating IDS for IoT-based smart environments.	It has lacking of adaptability to changing
Deep Learning for Intrusion Detection in IoT	[10]	Deep Learning for IoT Intrusion Detection	Leverages deep learning techniques for enhanced accuracy	Practical applicability in real world IoT environments is missing. Missing of integrating it in different IoT environments and ecosystems, so that the actual accuracy is absent
Enhancing IoT Network Security through Deep Learning-Powered IDS	[11]	DL (Random Forests, CNN, AUC for multiclass classification) for IoT Network Security	Proposes multiple deep learning models for IDS	Missing of ample evaluation of the trade-offs and scalability of it in IoT security. Also misses the practical implementation and how this model will perform in the wider IoT networks
IoT Malicious Traffic Classification Using Machine Learning	[12]	Uses subsampling and Machine learning model such as Random forest to enhance IoT malware detection	Offers high classification accuracy and introduces innovative approaches to IoT malware detection	Model interpretability and generalization to diverse datasets remain areas of concern
Malicious Traffic Detection in IoT and Local Networks Using Stacked Ensemble Classifier	[13]	This research uses the Extra Boosting Forest(EBF) ensemble model to overcome the limitation while dealing with multi domain data	It is effective in detecting malicious traffic and the approach is versatile because it addresses both binary and multi class classification tasks	The EBF model is computationally complex and resource-intensive. This focuses in performance in experimental environment and has lackings of real-life implementation

3. Methodology

3.1. Data Collection

One of the major research challenges in this field is the unavailability of a comprehensive network based data set which can reflect modern network traffic scenarios, vast varieties of low footprint intrusions and depth structured information about the network traffic. Evaluating network intrusion detection systems research efforts, KDD98, KDDCUP99 and NSLKDD benchmark data sets were generated a decade ago. However, numerous current studies showed that for the current network threat environment, these data sets do not inclusively reflect network traffic and modern low footprint attacks. Countering the unavailability of network benchmark data set challenges, this paper examines a UNSW-NB15 data set creation. This data set has a hybrid of the real modern normal and the contemporary synthesized attack activities of the network traffic. Existing and novel methods are utilised to generate the features of the UNSWNB15 data set. This dataset have 49 features with the class label. These features are described in UNSW-NB15_features.csv file

3.2. Data preprocessing

The data cleaning, organizing, visualizing and finally, handling of the questions have been described exhaustively in this section.

3.2.1. Data cleaning

Our raw UNSW-NB15 dataset had 45 attributes and 175341 rows in it. It had many null values so we implement isnull() function [data.isnull().sum()] to figure out those null values. We got the result there were 94168 null values. To deduct those null values we used dropna() function. After dropping those null values Dataset had 45 attributes and 81173 rows. Data type of attributes is converted using provided datatype information from features dataset. The duplicate data checking and the feature-renaming process were conducted. Label and one-hot encodings were applied to provide a numerical representation of the categorical features. One-hot-encoding Categorical Columns 'proto', 'service', 'state' are one-hot-encoded using pd.get_dummies() and these 3 attributes are removed afterwards. data_cat Dataframe had 19 attributes after one-hot-encoding. data_cat is concatenated with the main data dataframe. Total attributes of data dataframe is 61. Null valued entries were replaced with their corresponding mean values. We converted integer, binary, float columns to numeric. Min–max scaler was used to normalize features so that all numerical features have an acceptable range. In Data Normalization 58 Numeric Columns of DataFrame is scaled using MinMax Scaler.

3.2.2. Data visualization

For Binary classification we have created a copy of DataFrame. 'label' attribute is classified into two categories 'normal' and 'abnormal'. Now, by data visualization Figure 2 we get to see information on the percentage of normal labels and abnormal labels of the collected UNSWNB15 dataset. According to this figure, normal labels percentage is quite higher than the abnormal but labels but the abnormal labels are the most concerning factor here. We can see from the Pie Chart the normal labels percentage is 75.99 and the abnormal labels percentage is 24.01. The attribute 'label' is encoded using LabelEncoder(), encoded labels are saved in 'label'. In this Binary dataset we have 81173 rows and 61 columns. Same thing we did for the Multi-class classification. First we have created a copy the actual DataFrame for the Multi-class Classification. The attribute 'attack_cat' is classified into 9 categories which are 'Analysis', 'Backdoor', 'DoS', 'Exploits', 'Fuzzers', 'Generic', 'Normal', 'Reconnaissance', 'Worms'. The attribute attack_cat is encoded using LabelEncoder(), encoded labels are saved in label. attack_cat is also one-hot-encoded'. After that Multi-class Dataset have 81173 rows, 69 columns. From the Figure 3 we can the percentages of each categories.

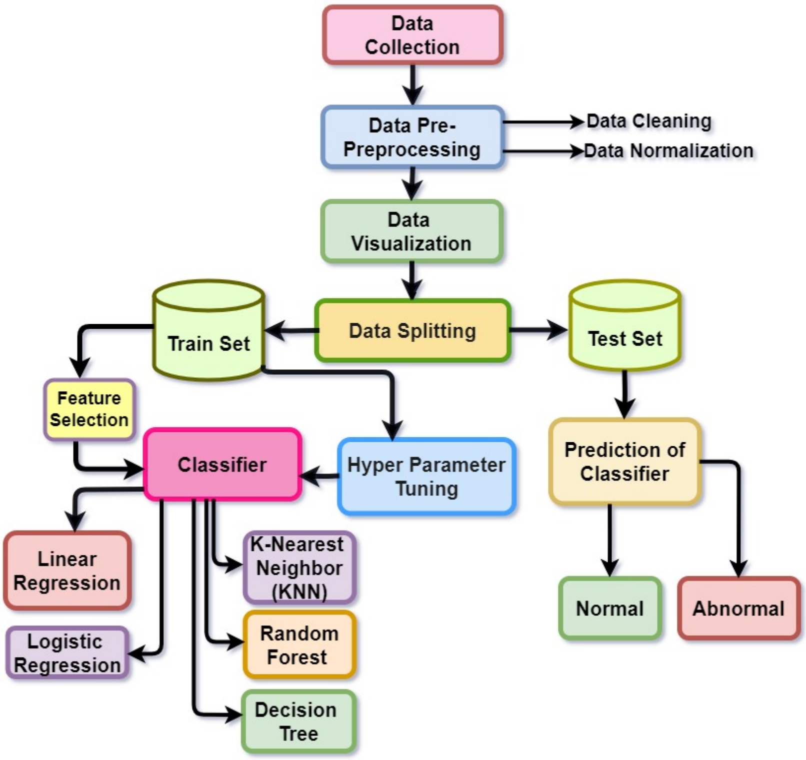


Figure 1: Methodology of the project

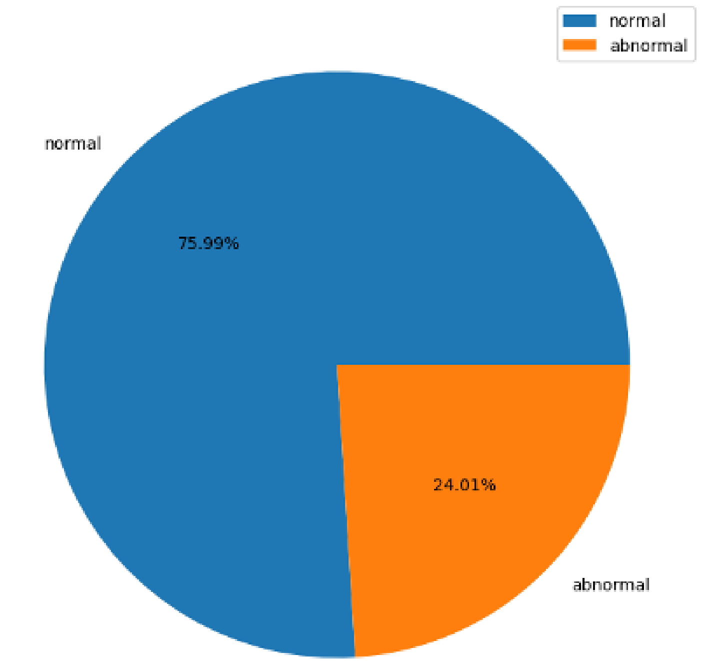


Figure 2: Pie chart distribution of normal and abnormal labels

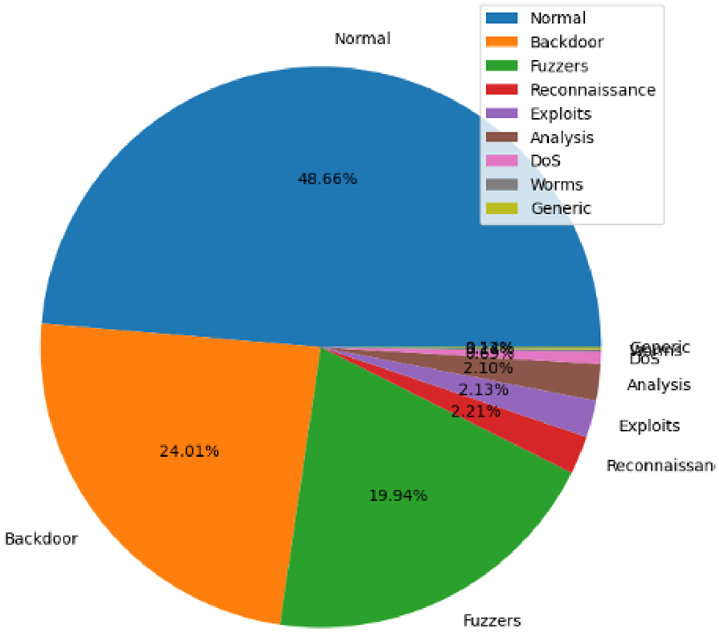


Figure 3: Pie chart distribution of multi-class labels

3.2.3. Feature Extraction

In 'bin_data' number of attributes are 61 and number of attributes of 'multi_data' is 69. We have implemented Pearson Correlation Coefficient method is used for feature extraction. The attributes with more than 0.3 correlation coefficient with the target attribute label were selected. After the feature extraction the attributes of 'bin_data' is 15('rate', 'sttl', 'sload', 'dload', 'ct_srv_src', 'ct_state_ttl', 'ct_dst_ltm', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm', 'ct_src_ltm', 'ct_srv_dst', 'state_CON', 'state_INT', 'label'). Number of attributes of 'multi_data' after feature selection is 16('dttl', 'swin', 'dwin', 'tcprtt', 'synack', 'ackdat', 'label', 'proto_tcp', 'proto_udp', 'service_dns', 'state_CON', 'state_FIN', 'attack_cat_Analysis', 'attack_cat_DoS', 'attack_cat_Exploits', 'attack_cat_Normal.) Then we saved the prepared dataset to disk.

3.2.4. Data Splitting

Data splitting is crucial in evaluating classification models, enabling us to assess their performance on new and unseen data. A training set and a testing set make up the two halves of the dataset. The training set is hidden until evaluation, and the model discovers patterns and relationships in the data. Stratified sampling is used to preserve the distribution of classes in the target variable across both sets, ensuring fairness and accuracy. This eliminates the possibility of biases resulting from imbalanced class distributions. We utilize the popular train_test_split function from the scikit-learn library to split the data, giving us the option to set a desired ratio of 80% for training and 20% for testing. Stratified sampling helps us gain an objective assessment of the model's performance and reduces the possibility of overfitting, which guarantees the model will function well when applied to fresh and untested data.

3.3. Classifier

The data is ready for the machine learning classifiers when the pre-processing and splitting stages are completed in the classifier stage. We assessed several classifiers in our work, including KNearest Neighbor, Linear Regression, Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines. For the best results, hyperparameter adjustment is needed for Decision trees, Random forests, and K-nearest neighbors classifiers. We used a three-fold cross-validation on the training set to make hyperparameter adjustments for each classifier to accomplish this.

- Linear Regression

  A fundamental and adaptable statistical technique, linear regression is important in many fields, such as data analysis, machine learning, economics, and the social sciences. By determining the linear equation that best captures this correlation, it is used to model and quantify the associations between a dependent variable (the target) and one or more independent variables (the predictors). One of the most important and often used tools for simulating linear connections between variables is linear regression. It facilitates hypothesis testing, permits well-informed forecasts, and offers insightful information about data trends. Because of its wide range of applications, it is a crucial method in machine learning and data analysis. Understanding and using linear regression is a vital first step in making sense of data and using it for predictions and decision-making in a variety of domains, despite its drawbacks.

- Logistic Regression
  A mathematical modeling method called logistic regression is used to explain the relationship between several independent variables, Z1–Zn, and a dependent variable, D. The logistic function, which for any given input has a range of 0 to 1, is used by the logistic model as a mathematical form. A probability of an occurrence, which is always a number between 0 and 1, can be described by the logistic model. The logistic model is represented by the following formula 1.

$$y = \alpha_0 + \alpha_1 Z_1 + \alpha_2 Z_2 + \ldots + \alpha_n Z_n \tag{1}$$

  Here, Z1, Z2,..., and y represent the response variables. The variable that is anticipated is Zn. The logistic function 2 is obtained by putting the 20 sigmoid functions to use.

$$l = \frac{1}{1 + e^{-(\alpha_0 + \alpha_1 Z_1 + \alpha_2 Z_2 + \ldots + \alpha_n Z_n)}} \tag{2}$$

- K-Nearest Neighbor (KNN)
  Regression and classification can be performed with the K-Nearest Neighbor technique, a straightforward, non-parametric supervised learning algorithm. All of the already-available examples are stored and newly found cases are categorized using feature similarity (such as the distance function). The KNN classification results in an output class member. The predominance vote of a case's neighbors determines its classification. Its K-nearest neighbors are assigned the case to the most common class. In the KNN approach, the value of K (positive integer) can be chosen using a variety of heuristic techniques. If K=1, the case will be placed in the nearest neighbor's class. The KNN method uses many distance functions, such as the Manhattan Distance, Euclidean Distance, and Minkowski Distance. This study makes use of the Minkowski Distance function. The following equation 3, where q denotes the Minkowski Distance's order, can be used to express the Minkowski Distance for two points: U (u1, u2,...., un) and V (v1, v2,...., vn).

$$\text{distance}(U, V) = \sum_{i=1}^{n} ((|u_i - v_i|)^q)) \tag{3}$$

- Random Forest (RF)

  By using a bootstrap aggregating technique, the Random Forest improves the performance of individual basic decision trees. To specifically train the various decision tree models, Random Forest bootstraps the original training data into several variants of the training set. During the classification process, a voting mechanism including all simple decision tree classifiers forms the conclusion for each new instance that needs to be classified.

- Decision Tree
  A decision tree is a model that describes the potential states that could correlate to events using a tree structure. Decision trees are a non-parametric supervised learning technique commonly used in machine learning to address regression and classification issues. Learning to create a decision tree involves breaking the training into smaller subsets, each of which should be as "pure" as feasible. The number of training components with the same class label in a given collection indicates its purity. In actuality, the algorithms Hunts, ID3, C4.5, and J48—an implementation of C4.5 found in Weka software 19—are used to build decision trees. The idea of information gains and entropy serves as the foundation for this algorithm.

$$E(S) = X - \sum_{i=1}^{n} p_i \log_2(p_i)c_i = 1 \tag{4}$$

- Hyper-parameter tuning
  The process of finding the optimal model architecture through adjustments to hyperparameters is known as hyperparameter tuning. The model's learning process is aided by these hyperparameters, whose values are established before the model's initialization. The simplest technique for hyperparameter tuning is probably grid search. This method involves creating a model for every possible combination of the given hyperparameter values, assessing each model, and choosing the architecture that yields the best results. By providing a statistical distribution for each hyperparameter from which values may be randomly picked, random search departs from grid search in that it no longer provides a discrete set of values to explore for each hyperparameter.

4. Results

To test and assess the suggested model, this section provides more details on the classifiers used. The Mean Absolute Error and Root Mean Squared Error have been found for every model and mainly the precision, recall and f-1 score have been found to evaluate and compare the models' performances for the predictions.

The Mean Absolute Error (MAE) is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where:

n is the number of data points,
y_i is the observed (actual) value for data point i,
$\hat{y}_i$ is the predicted value for data point i.

The Root Mean Squared Error (RMSE) is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where:

n is the number of data points,
y_i is the observed (actual) value for data point i,
$\hat{y}_i$ is the predicted value for data point i.

The Precision is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

where:

TP is the number of true positives,
FP is the number of false positives.

The Recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

where:

TP is the number of true positives,
FN is the number of false negatives.

The F1-Score is defined as the harmonic mean of Precision and Recall:

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Correlation matrices for binary and multi-class classification have been included in this section, followed by visualizations comparing real and predicted values for the different models, which consist of confusion matrices and tables of classification reports. The tables compare precision, recall f-1 scores of different models.

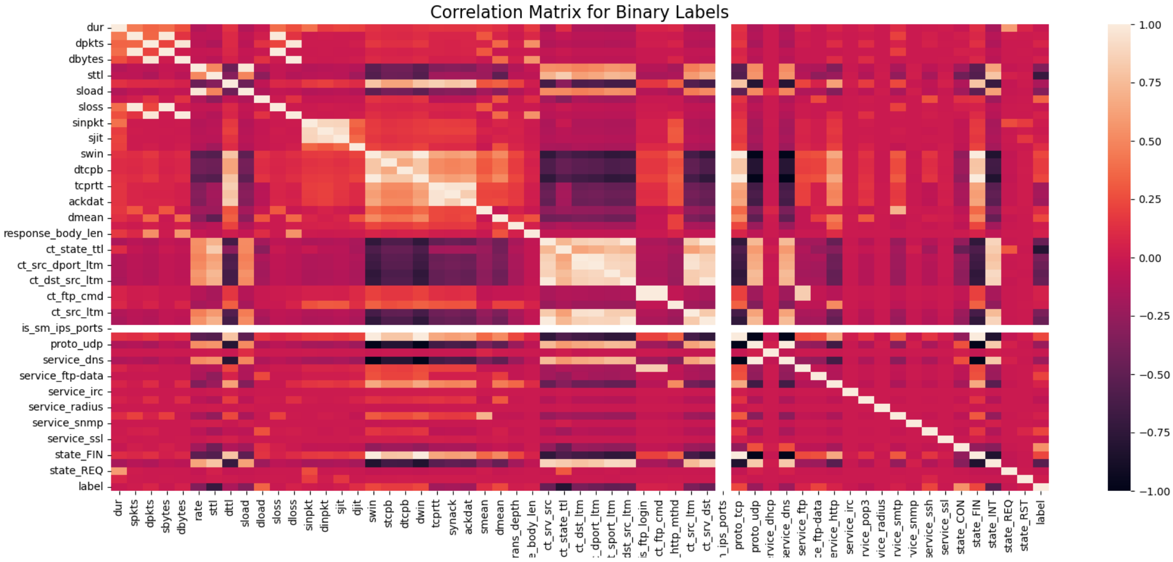


Figure 4: Correlation Matrix for Binary Labels

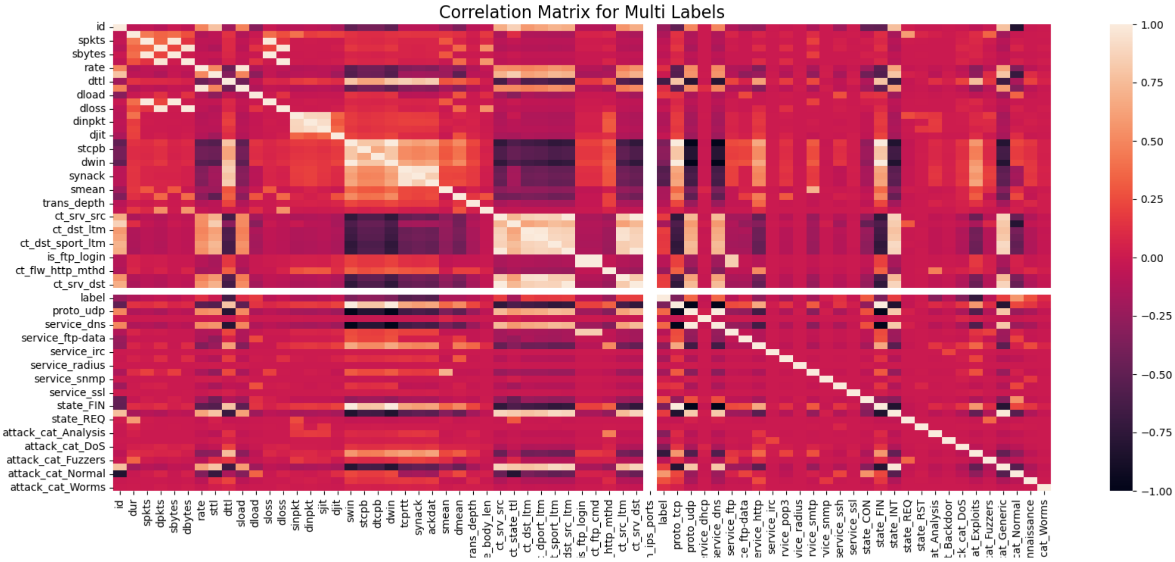


Figure 5: Correlation Matrix for Multi-class Labels

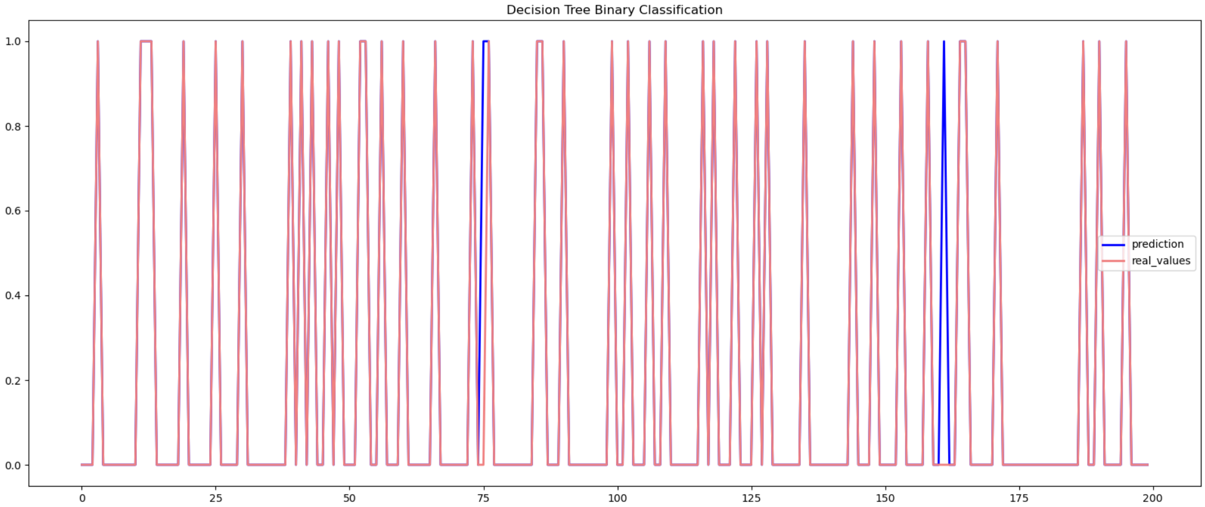


Figure 6: Decision Tree classification for binary labels

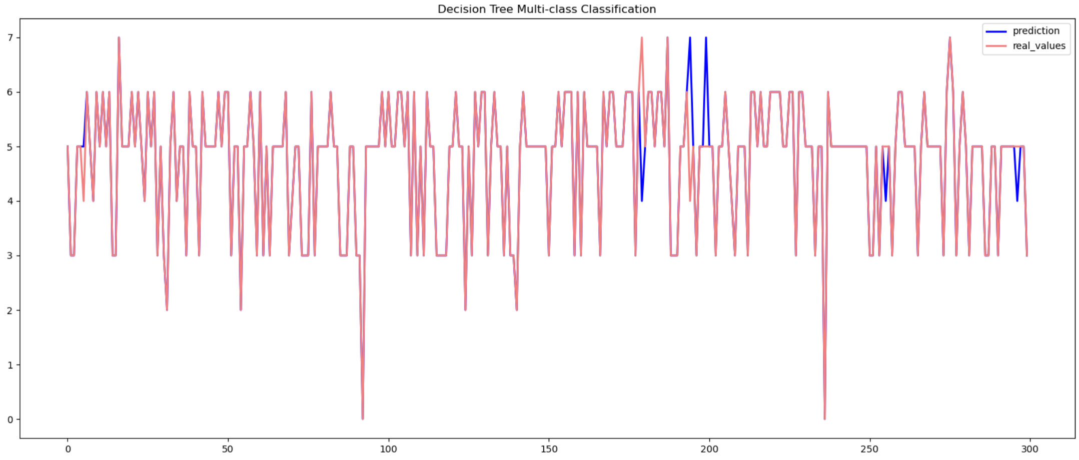


Figure 7: Decision Tree classification for multi-class labels

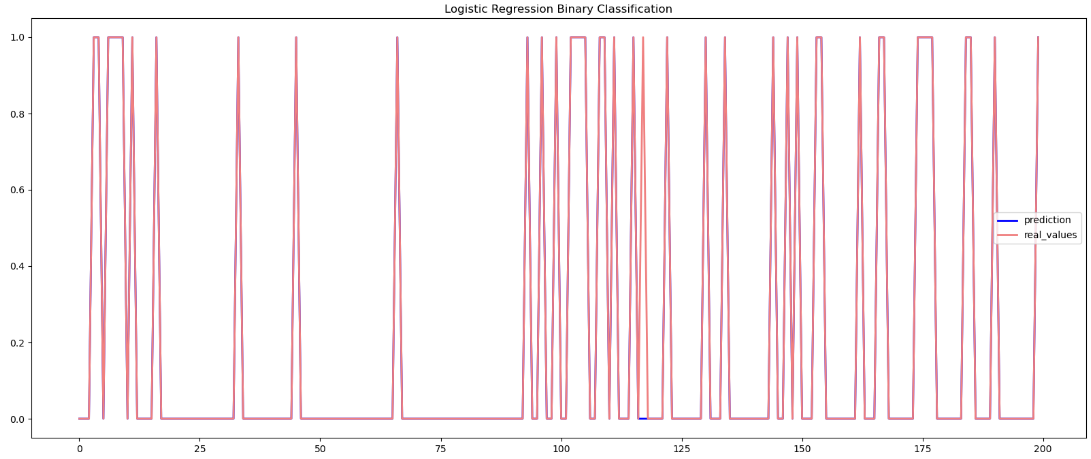


Figure 8: Logistic Regression Classification (Binary)

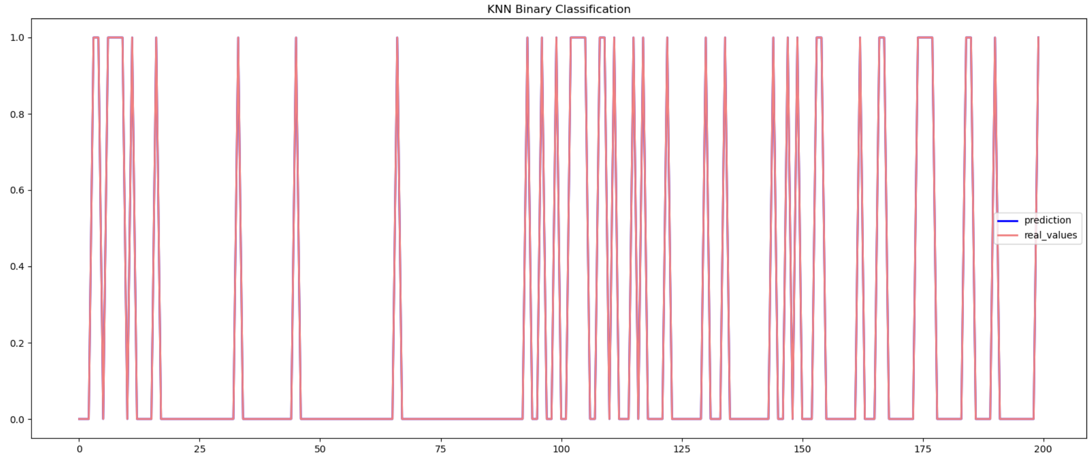


Figure 9: KNN Classification (Binary)

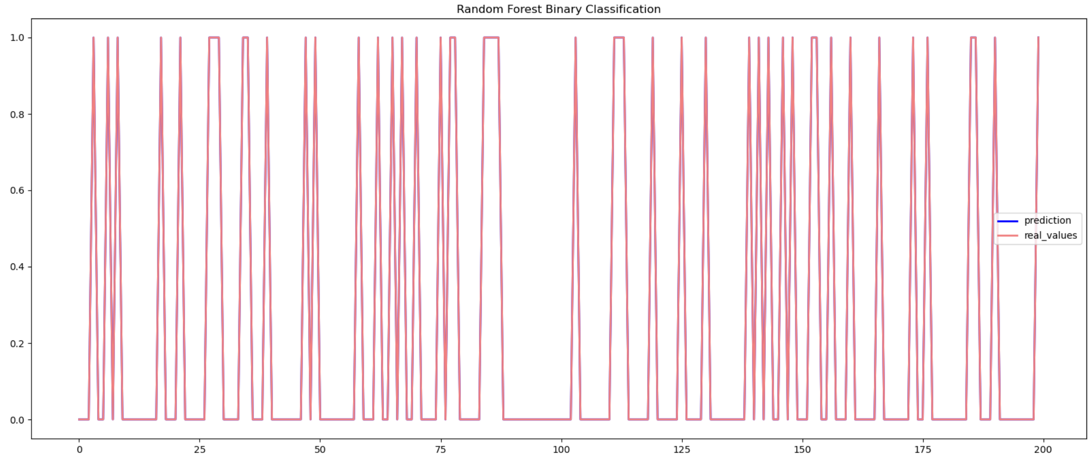
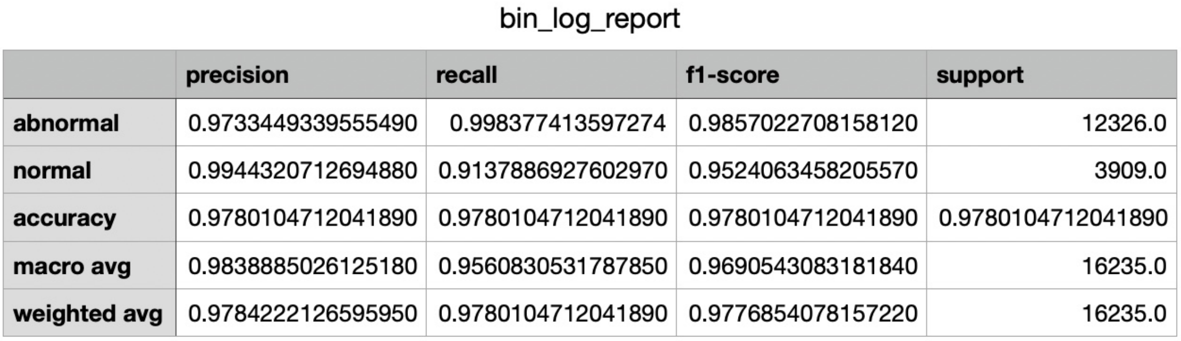


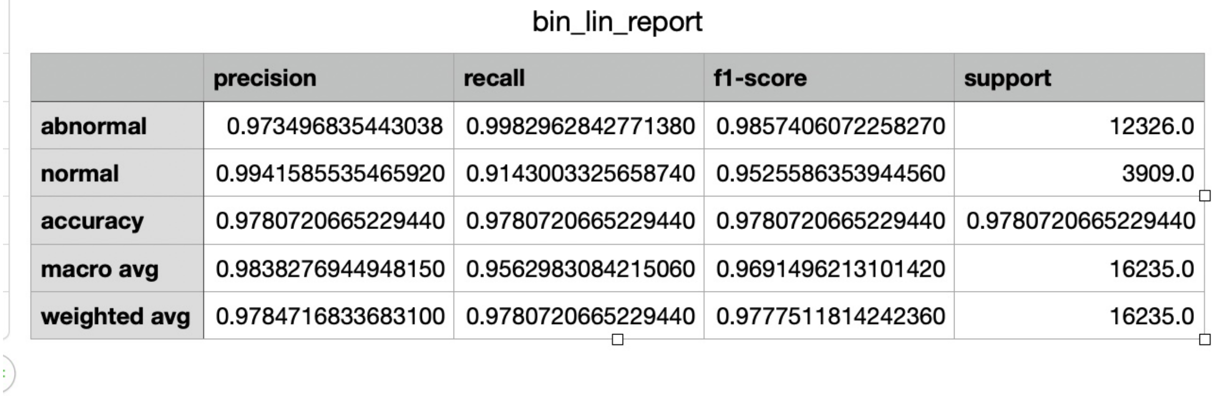
Figure 10: Random Forest Classification (Binary)



bin_log_report

	precision	recall	f1-score	support
abnormal	0.9733449339555490	0.998377413597274	0.9857022708158120	12326.0
normal	0.9944320712694880	0.9137886927602970	0.9524063458205570	3909.0
accuracy	0.9780104712041890	0.9780104712041890	0.9780104712041890	0.9780104712041890
macro avg	0.9838885026125180	0.9560830531787850	0.9690543083181840	16235.0
weighted avg	0.9784222126595950	0.9780104712041890	0.9776854078157220	16235.0

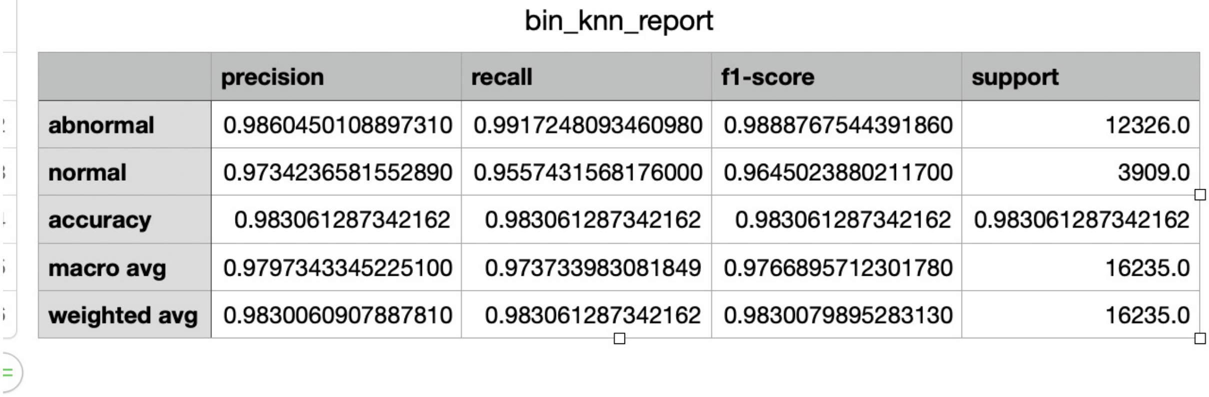
Figure 11: Class report for Logistic Regression



bin_lin_report

	precision	recall	f1-score	support
abnormal	0.973496835443038	0.9982962842771380	0.9857406072258270	12326.0
normal	0.9941585535465920	0.9143003325658740	0.9525586353944560	3909.0
accuracy	0.9780720665229440	0.9780720665229440	0.9780720665229440	0.9780720665229440
macro avg	0.9838276944948150	0.9562983084215060	0.9691496213101420	16235.0
weighted avg	0.9784716833683100	0.9780720665229440	0.9777511814242360	16235.0

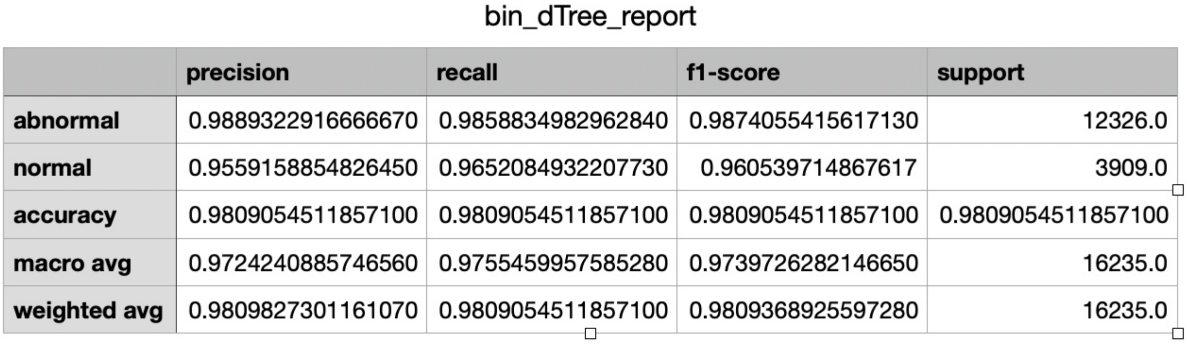
Figure 12: Class report for Linear Regression



bin_knn_report

	precision	recall	f1-score	support
abnormal	0.9860450108897310	0.9917248093460980	0.9888767544391860	12326.0
normal	0.9734236581552890	0.9557431568176000	0.9645023880211700	3909.0
accuracy	0.983061287342162	0.983061287342162	0.983061287342162	0.983061287342162
macro avg	0.9797343345225100	0.973733983081849	0.9766895712301780	16235.0
weighted avg	0.9830060907887810	0.983061287342162	0.9830079895283130	16235.0

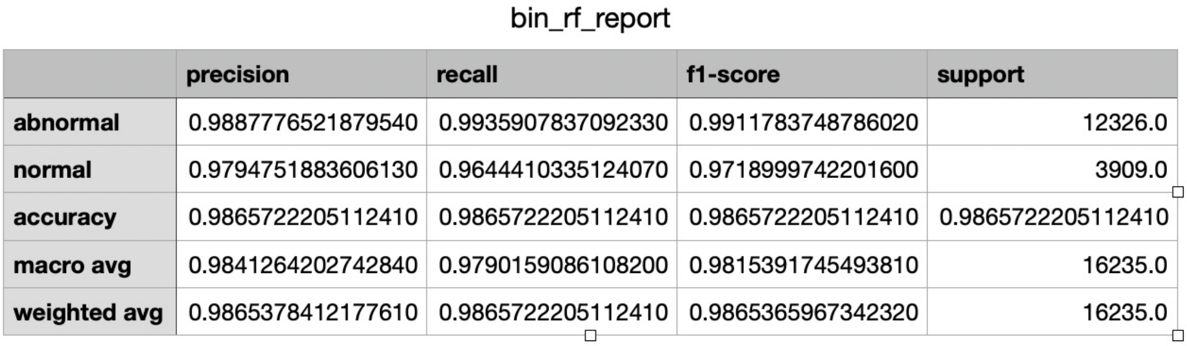
Figure 13: Class report for KNN



bin_dTree_report

	precision	recall	f1-score	support
abnormal	0.9889322916666670	0.9858834982962840	0.9874055415617130	12326.0
normal	0.9559158854826450	0.9652084932207730	0.960539714867617	3909.0
accuracy	0.9809054511857100	0.9809054511857100	0.9809054511857100	0.9809054511857100
macro avg	0.9724240885746560	0.9755459957585280	0.9739726282146650	16235.0
weighted avg	0.9809827301161070	0.9809054511857100	0.9809368925597280	16235.0

Figure 14: Class report for Decision Tree



bin_rf_report

	precision	recall	f1-score	support
abnormal	0.9887776521879540	0.9935907837092330	0.9911783748786020	12326.0
normal	0.9794751883606130	0.9644410335124070	0.9718999742201600	3909.0
accuracy	0.9865722205112410	0.9865722205112410	0.9865722205112410	0.9865722205112410
macro avg	0.9841264202742840	0.9790159086108200	0.9815391745493810	16235.0
weighted avg	0.9865378412177610	0.9865722205112410	0.9865365967342320	16235.0

Figure 15: Class report for Random Forest

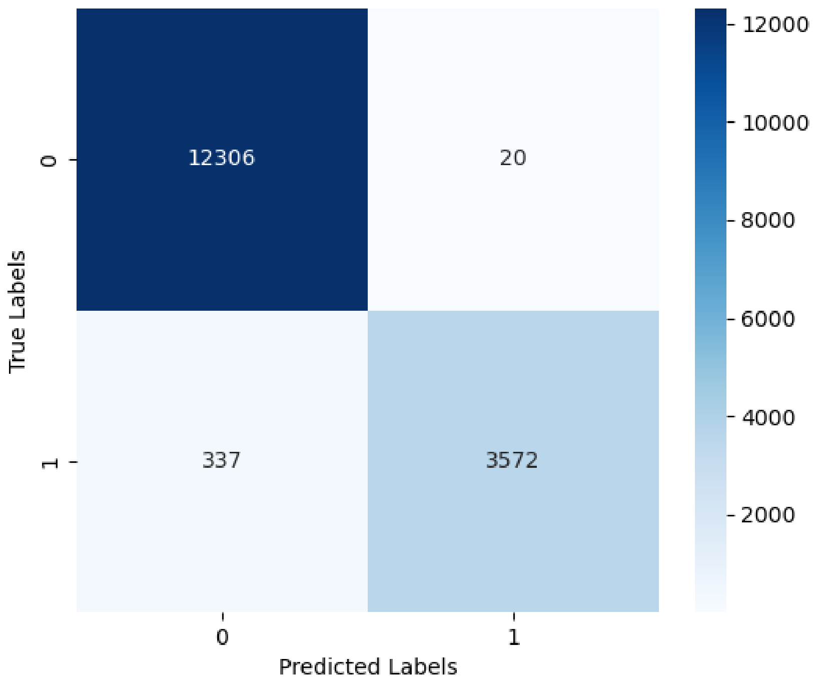


Figure 16: Confusion Matrix for Logistic Regression

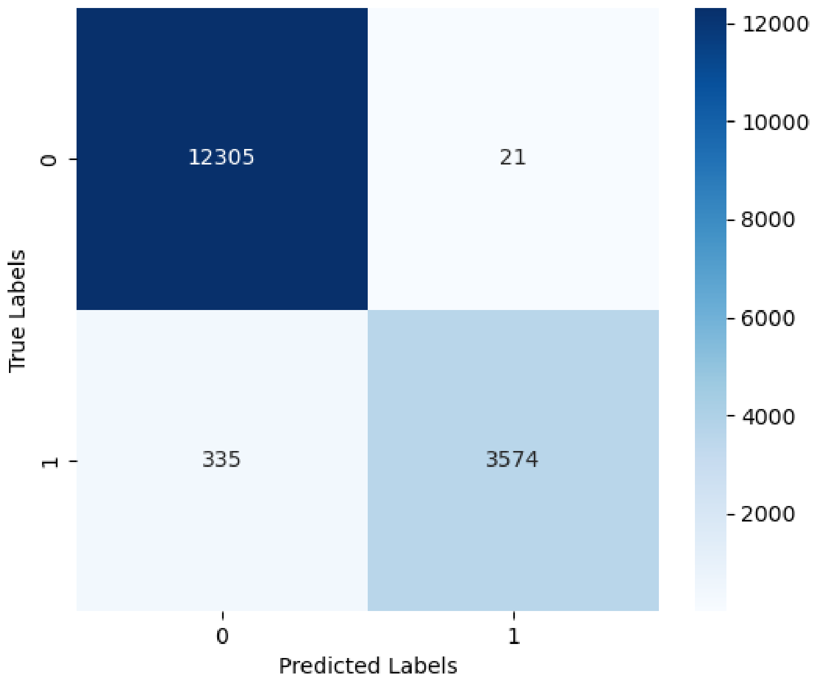


Figure 17: Confusion Matrix for Linear Regression

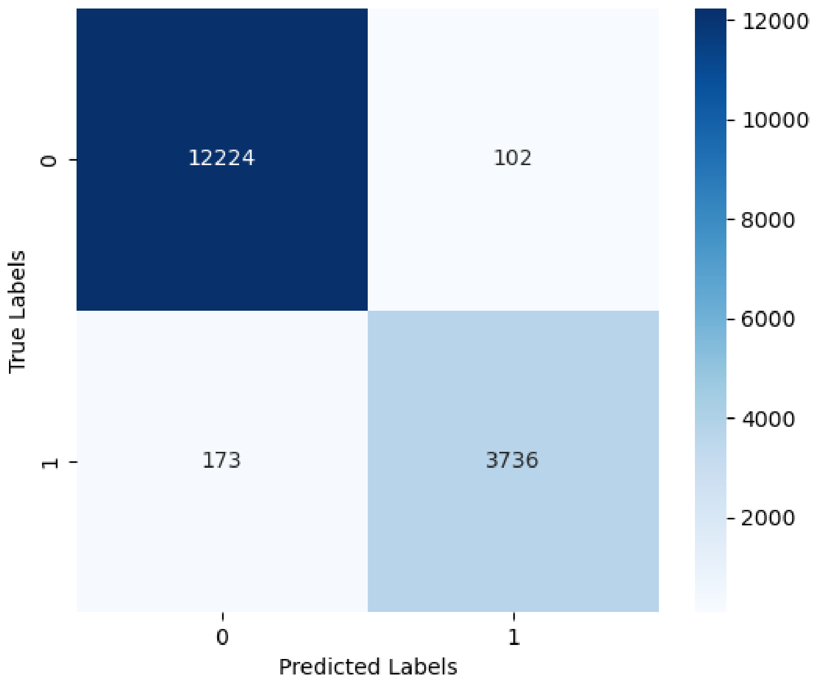


Figure 18: Confusion Matrix for KNN

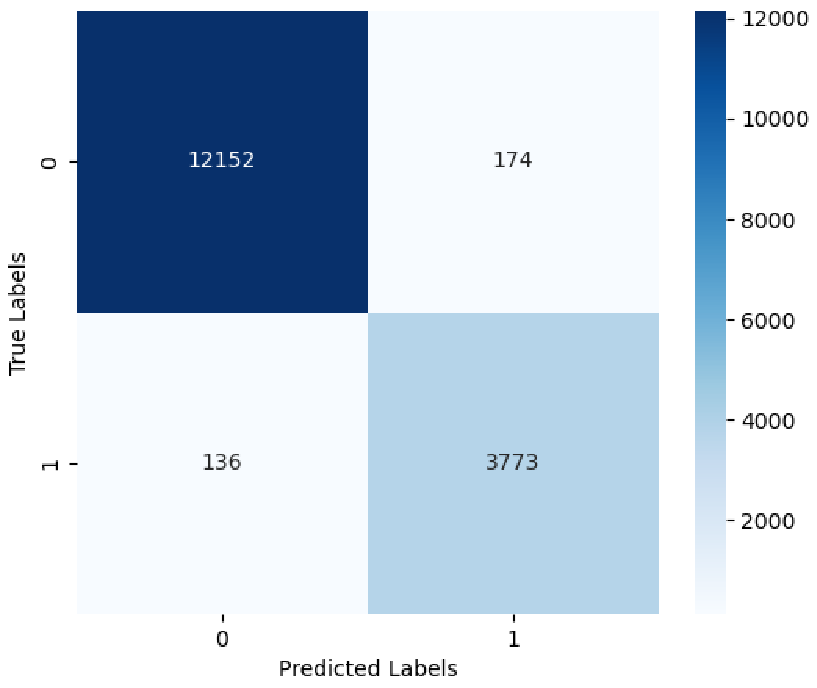


Figure 19: Confusion Matrix for Decision Tree

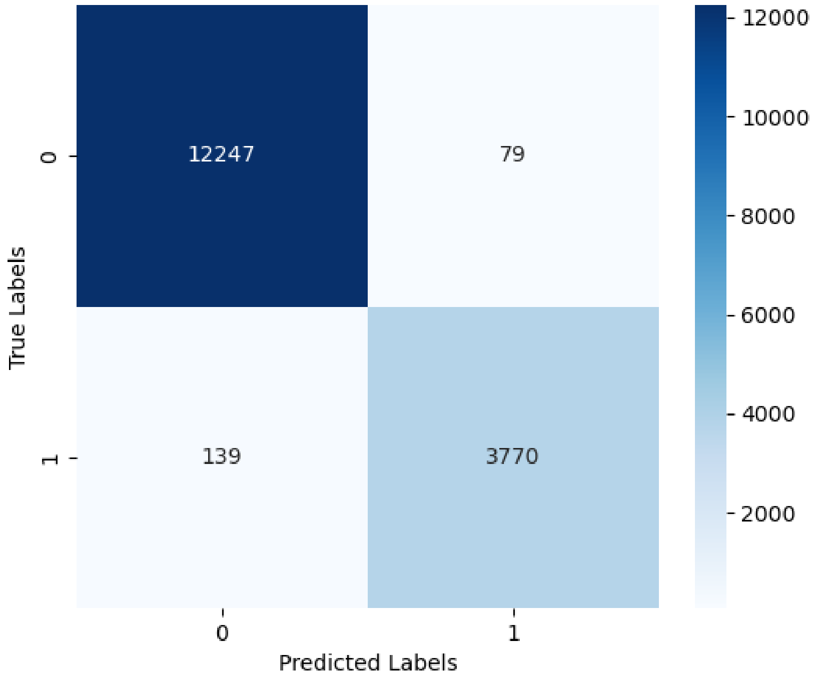


Figure 20: Confusion Matrix for Random Forest

5. Discussion

This research presents an essential contribution to the field of IoT security, especially for the development of IDS for IoT-based smart environments. The research paper thoroughly surveys and analyzes the latest IDS designed for the IoT environment. This research paper is an ideal guide for professionals, researchers, and businesses seeking ways to enhance the security of their IoT devices as it tackles the core issues of safeguarding systems in smart environments.

Compared to the other notable works in the field, the comprehensive approach of this survey stands out. While other studies have investigated specific areas of IoT security, This study gives a comprehensive analysis of IDS, providing a deep understanding of their methodologies and features. The focus on security vulnerabilities and the need for a specialized IDS highlights the significance of this research in handling the particular difficulties faced by IoT-based smart environments. In contrast to other prior research that may focus on specific areas of IoT security, this research gives a comprehensive view that might lead to creating more robust and effective security solutions. As a result, this research is a helpful resource for anyone looking to strengthen their IoT security procedures. It is a thorough guide to a constantly shifting and vital issue.

The result of this research presents the way to increase IoT security, especially in the field of IoT based smart environment using Intrusion Detection Systems. As an ideal guide, the outcome of this research will help the professionals, companies, researchers and anyone who wants to increase the system security by using the IDS.

6. Limitations

An essential component of making sure Internet of Things (IoT) networks and devices are secure is detecting illegal IoT network traffic. This field does, however, have certain restrictions and difficulties. A few noteworthy ones are as follows:

- Heterogeneity of IoT Devices:
  The Internet of Things (IoT) comprises a diverse array of devices with differing functionalities, methods for communication, and architectural designs. It is difficult to provide a universally applicable method for identifying malicious communications.

- Scalability:
  The volume of network traffic rises together with the number of IoT devices. When it comes to real-time processing and analysis of traffic data from several devices, scalability is a major barrier.

- Resource Limitations:
  IoT devices frequently have little amounts of memory, processing power, and energy. One limitation is the need to put strong security measures in place on small devices with limited resources without compromising their performance.

- Data privacy:
  Gathering and analyzing network traffic for malicious activity while protecting user privacy and adhering to data protection standards can be difficult due to the sensitive nature of IoT data and privacy concerns.

- Class Imbalance:
  Because fraudulent traffic frequently belongs to a minority class in the dataset, class imbalance might be problematic for machine learning-based detection. Prejudicial or imprecise models may result from imbalanced datasets.

- Regulatory Compliance:
  IoT networks need to abide by certain rules, especially in vital industries like healthcare and transportation. It might be challenging to meet these compliance standards while maintaining security.

- Lack of Standardization:
  It may be difficult to set up consistent security measures if there are no established security standards and practices for Internet of Things devices and networks.

It takes a multidisciplinary strategy that combines domain-specific understanding of IoT applications, data analysis, cybersecurity, and machine learning skills to address these obstacles and limits in the identification of harmful IoT network traffic. Moreover, continuous research and cooperation are necessary to keep up with the rapidly expanding IoT ecosystems and the changing threat landscape.

7. Conclusion and future work

This paper presents various machine learning algorithms whose performances have been carefully evaluated by us. Our findings have highlighted the varying effectiveness of these algorithms in real-world IoT network intrusion scenarios. Decision tree and random forest exhibited strong performance, whereas linear regression and logistic regression showed limited efficacy. KNN performed moderately well, indicating its potential in specific contexts. The insights gained from this comparative analysis provide valuable guidance for practitioners in selecting suitable intrusion detection techniques for IoT networks. In the future, we plan to Investigate ensemble techniques and deep learning architectures to enhance the robustness of intrusion detection models, especially in the face of evolving and sophisticated cyber threats. Simultaneously explore real-time implementation of efficient algorithms, ensuring low latency and high accuracy, critical for timely response to IoT security incidents.

8. References

1. D. Rani and N. C. Kaushal, ”Supervised Machine Learning Based Network Intrusion Detection System for Internet of Things,” 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225340.

2. Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, Hicham Lakhlef, Internet of things security: A top-down survey, Computer Networks, Volume 141, 2018, Pages 199-221, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2018.03.012.

3. C. Cyrus, “IOT cyberattacks escalate in 2021, according to Kaspersky,” IoT Cyberattacks Escalate in 2021, According to Kaspersky, https://www.iotworldtoday.com/security/iot-cyberattacks-escalate-in-2021-according-to-kaspersky

4. L. Phillips, “Healthcare cyberattacks increasing in 2023,” Insider Intelligence, https://www.insiderintelligence.com/content/healthcare-cybersecurity-2023-hive-s-shutdown-good-news-cyberattacks-only-getting-worse

5. “2022 Cyber Security Statistics Trends & Data,” PurpleSec. https://purplesec.us/resources/cyber-security-statistics/#IoT

6. “Why attackers love to target IoT devices,” VentureBeat, Jun. 09, 2023. https://venturebeat.com/security/why-attackers-love-to-target-iot-devices/

7. D. Stiawan, M. Y. Idris, R. F. Malik, S. Nurmaini and R. Budiarto, ”Anomaly detection and monitoring in Internet of Things communication,” 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Yogyakarta, Indonesia, 2016, pp. 1-4, doi: 10.1109/ICITEED.2016.7863271.

8. M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, “Intrusion detection systems for IoT-based smart environments: a survey,” Journal of Cloud Computing, vol. 7, no. 1, Dec. 2018, doi: https://doi.org/10.1186/s13677-018-0123-6.

9. A. Diro, N. Chilamkurti, V.-D. Nguyen, and W. Heyne, “A Comprehensive Study of Anomaly Detection Schemes in IoT Networks Using Machine Learning Algorithms,” Sensors, vol. 21, no. 24, p. 8320, Dec. 2021, doi: 10.3390/s21248320.

10. B. Susilo and R. F. Sari, “Intrusion Detection in IoT Networks Using Deep Learning Algorithm,” Information, vol. 11, no. 5, p. 279, May 2020, doi: 10.3390/info11050279.

11. S. A. Bakhsh, M. A. Khan, F. Ahmed, M. S. Alshehri, H. Ali, and J. Ahmad, “Enhancing IoT network security through deep learning-powered Intrusion Detection System,” Internet of Things, vol. 24, p. 100936, Dec. 2023, doi: https://doi.org/10.1016/j.iot.2023.100936.

12. H. Fowdur, S. Armoogum, G. Suddul and V. Armoogum, ”Detecting Malicious IoT Traffic using Supervised Machine Learning Algorithms,” 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2022, pp. 209-213, doi: 10.1109/ZINC55034.2022.9840635.

13. O. Olayemi Petinrin, F. Saeed, X. Li, F. Ghabban, and K.-C. Wong, “Malicious Traffic Detection in IoT and Local Networks Using Stacked Ensemble Classifier,” Computers, Materials & Continua, vol. 71, no. 1, pp. 489–515, 2022, doi: https://doi.org/10.32604/cmc.2022.019636.