iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.70003

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Predictive Modelling for Network Threat Detection Using Artificial Intelligence Techniques

Dr. R. Bharathi, Asha P V, M S Arjun Dev, M A Navin Raj
Department of Computer Science and Engineering, Department of Computer Science and Engineering with specialization in Cyber Security, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu

Abstract: ***The advent of artificial intelligence (AI) techniques has revolutionized network security by enabling predictive modelling for threat detection. This abstract proposes a novel approach to enhancing network security through predictive modelling, leveraging advanced AI techniques. By analyzing vast amounts of network traffic data, AI algorithms can identify patterns indicative of potential threats, including malware, intrusions, and anomalous activities. The predictive models developed through this approach can forecast potential network vulnerabilities and pre-emptively detect emerging threats before they manifest into security breaches. This proactive stance empowers organizations to fortify their network defenses, minimize the risk of cyberattacks, and safeguard sensitive information. Through the fusion of AI and predictive modelling, this research endeavors to pave the way for more robust and resilient network security frameworks in an increasingly interconnected digital landscape. Future improvements will focus on incorporating AI-driven adaptive security mechanisms that can evolve with emerging cyber threats. With the increasing reliance on digital platforms, this study highlights the urgent need for a comprehensive cybersecurity framework to safeguard business websites. The project not only presents a novel security solution but also provides insights into best practices for website protection, ensuring a safer digital environment.***

Keywords: Predictive modelling, Network security, Artificial intelligence, Threat detection, Cybersecurity, Machine Learning, Anomaly detection, Predictive analytics, Network traffic analysis, Cyber threats.

I. INTRODUCTION

With the rapidly evolving digital world, the increasing development of interconnected systems has come with a corresponding increase in vulnerability to cyber-attacks and network risk. Traditional threat detection methods of networks, normally reactive and based on signatures, cannot handle the dynamic and advanced nature of current cyber-attacks. This increasing disconnect has prompted the development of advanced technologies, such as artificial intelligence (AI), to come up with adaptive and proactive defense mechanisms. Among them, predictive modelling is an extremely promising option that offers the ability to predict potential threats and detect anomalies before they mature into significant security intrusions.

AI and machine learning-based predictive modelling enables systems to learn from volumes of network traffic data, picking up subtle patterns that may hint at malicious activities. By continuous monitoring in real-time, AI models can alert deviations from predicted norms, which can be signs of malware infection, attempts at unauthorized access, or zero-day attacks.

While traditional systems rely on predefined rules, predictive models can learn to accommodate the network environment, thereby being more successful against new and unknown attacks. The paper relies on AI-based predictive modelling and its deployment for enhanced threat detection in the network. The paper explores how various machine learning approaches, like anomaly detection, pattern analysis, and behavioral analysis, are integrated together to make predictions about possible vulnerabilities.

The goal is to develop an active network security environment that not only detects threats but also provides warnings ahead of time so that organizations can counter risks in real time. With the powers of predictive analytics, this project aims to construct stronger cybersecurity infrastructures and empower smarter, self-learning network defense systems.

A. Data Science

Data science is an interdisciplinary field that systematically applies scientific methods, algorithms, processes, and systems to extract knowledge and actionable insights from both structured and unstructured data, enabling their application across a wide range of disciplines. The origins of the term "data science" can be traced to 1974, when Peter Naur proposed it as an alternative designation for computer science. In 1996, the International Federation of Classification Societies convened the first conference that explicitly featured data science as a central theme, although its definition at that time remained unsettled. Within a short span, data science emerged as one of the most prominent and rapidly growing professional fields worldwide.

©IJRASET: All Rights are Reserved

Fundamentally, data science encompasses the integration of domain expertise, programming proficiency, and a robust foundation in mathematics and statistics to derive meaningful insights from data. It constitutes a synthesis of mathematical modeling, business intelligence, computational tools, algorithmic design, and machine learning methodologies, all aimed at uncovering latent patterns within raw datasets.

These insights are pivotal in informing strategic business decisions and advancing data-driven innovation.

B. Artificial Intelligence

Artificial intelligence (AI) refers to the simulation of mortal intelligence in machines that are programmed to suppose like humans and mimic their conduct. The term may also be applied to any machine that exhibits traits associated with a mortal mind similar as learning and problem- working.

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by humans or creatures. Leading AI handbooks define the field as the study of" intelligent agents" any system that perceives its terrain and takes conduct that maximize its chance of achieving its pretensions. Some popular accounts use the term" artificial intelligence" to describe machines that mimic" cognitive" functions that humans associate with the mortal mind, similar as" learning" and" problem working", still this description is rejected by major AI experimenters.

Artificial intelligence is the simulation of mortal intelligence processes by machines, especially computer systems. Specific operations of AI include expert systems, natural language processing, speech recognition and machine vision.

AI operations include advanced web hunt machines, recommendation systems(used by YouTube, Amazon and Netflix), Understanding mortal speech(similar as Siri or Alexa), tone- driving buses (e.g. Tesla), and contending at the loftiest position in strategic game systems(similar as chess and Go), As machines come decreasingly able, tasks considered to bear" intelligence" are frequently removed from the description of AI, a miracle known as the AI effect. For case, optic character recognition is constantly barred from effects considered to be AI, having come a routine technology.

Artificial intelligence was innovated as an academic discipline in 1956, and in the times since has endured several swells of sanguinity, followed by disappointment and the loss of backing (known as an" AI downtime"), followed by new approaches, success and renewed backing. AI exploration has tried and discarded numerous different approaches during its continuance, including bluffing the brain, modelling mortal problem working, formal sense, large databases of knowledge and imitating beast geste . In the first decades of the 21st century, largely fine statistical machine learning has dominated the field, and this fashion has proved largely successful, helping to break numerous gruelling problems throughout assiduity and academia. The colourful sub-fields of AI exploration are centred on pretensions and the use of tools. The traditional pretensions of AI disquisition include sense, knowledge representation, planning, Learning, natural language processing, perception and the capability to move and manipulate objects.

General intelligence (the capability to break an arbitrary problem) is among the field's long- term pretensions. To break these problems, AI researchers use performances of quest and fine optimization, formal sense, artificial neural networks, and styles predicated on statistics, probability and economics. As the hype around AI has accelerated, merchandisers have been scrambling to promote how their products and services use AI. constantly what they relate to as AI is simply one element of AI, analogous as machine Learning. No bone programming language is synonymous with AI, but a numerous, including Python, R and Java, are popular. In general, AI systems work by ingesting large amounts of labelled training data, assaying the data for correlations and patterns, and using these patterns to make prognostications about future countries. In this way, a chatbot that is fed samples of text exchanges can learn to produce life like exchanges with people, or an image recognition tool can learn to identify and describe objects in images by reviewing millions of samples. AI is important because it can give enterprises perceptivity into their operations that they may not have been alive of previously and because, in some cases, AI can perform tasks better than humans.

Particularly when it comes to repetitive, detail- acquainted tasks like assaying large numbers of legal documents to ensure applicable fields are filled in properly, AI tools constantly complete jobs snappily and with fairly numerous crimes. Artificial neural networks and deep Learning artificial intelligence technologies are snappily evolving, primarily because AI processes large amounts of data important hastily and makes prognostications more directly than humanly possible.

C. Machine Learning

Machine Learning is to predict the future from formerly data. Machine Learning (ML) is a type of artificial intelligence (AI) that provides computers with the capability to learn without being explicitly programmed. Machine Learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, performance of a simple machine learning algorithm using python.

The process of training and vacancy involves the use of specialized algorithms. It feeds the training data to an algorithm, and the algorithm uses this training data to give prognostications on new test data. Machine Learning can be roughly separated into three orders. There are supervised Learning, unsupervised Learning and underpinning Learning. Supervised Learning program is both given the input data and the corresponding labelling to learn data must be labelled by a mortal being beforehand. Unsupervised Learning is no labels. It handed the Learning algorithm. This algorithm must figure out the clustering of the data input. ultimately, bolstering Learning roundly interacts with its terrain, and it receives positive or negative feedback to meliorate its performance.

Data scientists use multitudinous kinds of machine Learning algorithms to discover patterns in python that lead to practicable perceptivity. At a high position, these different algorithms can be classified into two groups predicated on the way they "learn" about data to make prognostications supervised and unsupervised Learning. type is the process of predicting the class of given data points. Classes are sometimes called targets labels or orders. In machine Learning and statistics, type is a supervised Learning approach in which the computer program learns from the data input given to it and also uses this Learning to classify new obediences. This data set may simply bebi- class(like relating whether the person is virile or womanish or that the correspondence is spam ornon- spam) or it may bemulti- class too. Some samples of type problems are speech recognition, handwriting recognition, bio metric identification, document type etc. Supervised Machine Learning is the ultimate of the practical machine learning uses supervised Learning. Supervised Learning is where input variables(X) and an affair variable(y) and use an algorithm to learn the mapping function from the input to the affair is y = f(X). The thing is to compare the mapping function so well that when you have new data input(X) that you can predict the affair variables(y) for that data. ways of Supervised Machine Learning algorithms include logistic regression, Ulti- class type, Decision Trees and support vector machines etc. Supervised Learning requires that the data used to train the algorithm is formerly labelled with correct answers. Supervised Learning problems can be further grouped into type problems. This problem has as thing the construction of a brief model that can predict the value of the dependent particularity from the particularity variables. A type model attempts to draw some conclusion from observed values. Given one or farther inputs a type of model will try to predict the value of one or farther issues. A type of problem is when the affair variable is an order, analogous as "red" or "blue".

Fig.1 Process of Machine Learning

II. RELATEZ WORK

This study highlights the significant advancements and existing methodologies in networks detection interconnected with Artificial Intelligence.

1) Adel Abusitta et al This study addresses the increasing complexity of malware and the challenges it poses for detection and classification. It highlights how current techniques help reverse engineer's analyses malware patterns and adapt to evolving threats. By incorporating novel composition analysis methods, the research provides deeper insight into malware behavior and attacker intent. A comprehensive survey of existing literature is presented, comparing classification approaches, evasion techniques, and feature extraction strategies. Each method is evaluated based on its algorithms and features, identifying strengths and limitations. The study concludes by outlining key challenges and suggesting future directions to improve malware analysis and detection.
2) R. Bharathi et al. (2022) examines sentiment analysis methods applied to Amazon unlocked mobile reviews using supervised learning models. The study incorporates text preprocessing techniques such as negation handling, punctuation elimination, stemming, and stop-word removal. To transform text data into numerical formats, feature extraction is carried out using the TF-IDF vectorizer. Three machine learning algorithms—Gaussian Naïve Bayes (GNB), Logistic Regression (LR), and Support Vector Machine (SVM)—are employed to categorize sentiments as positive, negative, or neutral. Validation experiments on Kaggle's benchmark dataset reveal that the SVM model surpasses the others, achieving superior accuracy, precision, recall, and F1-score. The results indicate that machine learning-driven sentiment classification significantly improves the analysis of customer feedback and the efficiency of product evaluations. Future research may investigate deep learning and clustering methodologies to further enhance sentiment classification outcomes.
3) The CNN serves as a low-resource, high-accuracy feature extractor, achieving 98.03% accuracy—surpassing models like VGG16, ResNet50, and InceptionV3. A hybrid CNN+SVM model further improves performance, replacing SoftMax with a Linear SVC classifier. The fine-tuned model attains 99.59% accuracy and faster execution, demonstrating the effectiveness of deep learning in scalable malware detection.

4) The research conducted by R. Bharathi et al. (2024) aims to improve sentiment analysis of online book reviews through the application of deep learning methodologies. It investigates the use of Convolutional Neural Networks (CNN) and Cascaded Recurrent Neural Networks (CRNN), in conjunction with word representation techniques such as N-grams and Global Vectors (GloVe), to enhance polarity classification. This study utilizes Amazon Kindle review datasets to explore the linguistic elements that affect the accuracy of sentiment classification. By employing GloVe embeddings for better contextual comprehension, the research combines CNN for feature extraction with CRNN for sequential analysis in predicting sentiment. The experimental findings indicate that the proposed model achieves a remarkable 98% improvement in sentiment classification accuracy, surpassing traditional methods.

5) Robert Chun et al This report explores the increasing use of Portable Document Format (PDF) files as a medium for cyber-attacks. With the rapid growth of technology, attackers exploit system vulnerability by embedding malware in PDFs due to their flexible structure. This makes PDFs an attractive target for spreading malicious content like worms and viruses. The report analyzes the inherent flexibility of PDF files that enables such exploits and examines why attackers favor this method. It also proposes the development of Python-based detection techniques to identify and prevent the spread of malicious PDFs, enhancing system and network security against evolving cyber threats.

6) R. Bharathi investigates optimization algorithms for analyzing bank loan histories, focusing on the Bat Algorithm, Particle Swarm Optimization (PSO), and Grey Wolf Optimization (GWO) to predict loan repayment behaviors. These algorithms are utilized on financial datasets to evaluate their effectiveness in understanding user behavior related to loan payments. The research assesses the advantages and disadvantages of each technique, emphasizing the adaptability of the Bat Algorithm, the efficiency of PSO, and the strategic methodology of GWO in loan prediction. The findings indicate that optimization techniques can significantly enhance financial forecasting, assisting banks in risk evaluation. However, the study also notes limitations such as computational complexity and challenges in adapting these methods for real-time use.

7) Akoh Atadoga et al A comprehensive review explores the role of machine learning (ML) in enhancing network security and threat detection. The study discusses various ML techniques, including supervised, unsupervised, and deep learning methods, highlighting their strengths and limitations in threat detection. The review identifies promising areas for further research, such as federated learning, adversarial machine learning, and explainable AI, emphasizing the potential of ML to strengthen network defenses against evolving cyber threats.

8) Sewak et al Reviews the application of deep reinforcement learning (DRL) in cybersecurity threat detection and protection. The research highlights how DRL algorithms have shown promise in developing AI-based solutions for complex cybersecurity challenges, including threat detection and endpoint protection. Unlike traditional supervised learning methods, DRL offers diverse applications, empowering innovative approaches in the threat defense landscape. The review fills a gap in literature by providing a comprehensive analysis of DRL's unique applications and accomplishments in cybersecurity, emphasizing its potential in augmenting systems with general AI capabilities for enhanced threat defense

9) Okoli et al presents the application of machine learning (ML) in cybersecurity, emphasizing its role in threat detection and defense mechanisms. The research highlights how ML algorithms can analyze extensive datasets to identify patterns and anomalies indicative of cyber threats. It discusses various ML methodologies, including supervised, unsupervised, deep learning, and reinforcement learning, assessing their suitability for different threat detection scenarios. The study also addresses challenges such as adversarial attacks, biased datasets, and the interpretability of ML models, underscoring the need for a holistic approach that integrates advanced technology with ethical considerations to enhance cybersecurity defense. Shone N. et al. introduces a novel deep learning-based intrusion detection system tailored for IoT environments using a combination of deep autoencoders and classical machine learning classifiers. Evaluated using the NSL-KDD dataset, the proposed model demonstrates high detection rates and superior accuracy compared to traditional ML algorithms. The study highlights that the integration of deep learning for feature extraction significantly improves the classification capability, especially in complex attack scenarios. However, the authors acknowledge the increased training time and the need for tuning hyperparameters, suggesting future work focus on optimization techniques for real-time application in IoT systems.

10) R. Bharathi et al. (2024) examines the application of deep learning techniques for sentiment analysis (SA) in online book reviews, particularly those found on Amazon Kindle. Traditional methods of sentiment analysis include lexicon-based and machine learning techniques; however, deep learning provides enhanced accuracy without the need for extensive feature engineering. This research presents an advanced deep learning sentiment classification model that integrates feature extraction methods such as N-grams and Global Vectors for Word Representation (GloVe). It employs cascaded recurrent neural networks (CRNN) and convolutional neural networks (CNN) for the classification of sentiments. Experimental results demonstrate the proposed GloVe-

CNN model's superior performance, achieving an accuracy of 98.00%, precision of 96.95%, recall of 96.13%, and an F-score of 96.52%.

III. PROPOSED WORK

The envisioned system for predictive modeling in network threat detection harnesses the power of artificial intelligence (AI) techniques to fortify cybersecurity infrastructure. By amalgamating advanced machine learning algorithms with comprehensive network data analysis, the proposed system aims to preemptively identify and neutralize potential threats before they infiltrate critical networks. Through continuous monitoring and analysis of network traffic patterns, anomaly detection algorithms can flag suspicious activities indicative of cyber threats such as malware infiltration, intrusions, and data breaches.

Moreover, by leveraging deep learning models trained on vast datasets of historical cyber incidents, the system can adapt and evolve its threat detection capabilities, staying ahead of emerging threats and evolving attack vectors. Furthermore, the integration of AI-driven predictive modeling enhances response times, allowing security teams to swiftly deploy countermeasures and mitigate potential damages. By proactively safeguarding network infrastructures, the proposed system serves as a pivotal defense mechanism against the ever-evolving landscape of cyber threats, ensuring robust cybersecurity posture and uninterrupted operations for organizations across various sectors.

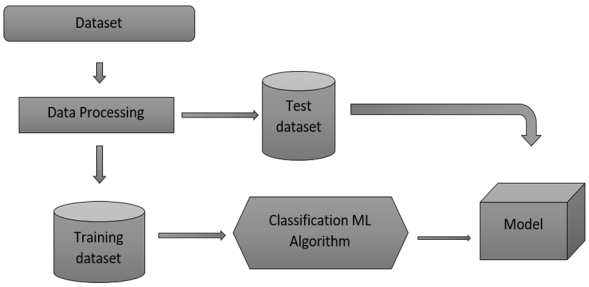


Fig.3 Architecture Diagram of the Proposed System

A. *Merits*

1) We use structured data for network traffic attack classification using advance machine learning methods.
2) We build a framework-based application for deployment purposes.
3) Accuracy was improved.
4) We classify more than 5 attacks.
5) We compared more than two algorithms to get better accuracy level.

B. *Existing System*

Furthermore, 150 cellular networks worldwide have rolled out LTE- M (LTE- Machine Type Communication) and or NB- IoT (Narrow Band Internet of goods) technologies to support massive IoT services analogous such as smart metering and environmental monitoring. analogous cellular IoT services partake the living cellular network architecture with non- IoT (e.g., smartphone) bones. In this work, we explore the security vulnerabilities of cellular IoT from both system- integrated and service- integrated aspects. We discovered several vulnerabilities gauging cellular standard design scars, network operation slips, and IoT device performance excrescencies. Threateningly, they allow an adversary to ever identify IP addresses and phone numbers assigned to cellular IoT bias, intrude their power saving services, and launch various attacks, including data text spamming, battery draining, device hibernation against them. The attack evaluation result shows that the adversary can raise an IoT data bill by over to$ 226 with lower than 120 MB spam business, increase an IoT text bill at a rate of$ 5 per second, and help an IoT device from entering leaving power saving mode; also, cellular IoT bias may suffer from denial of IoT services. We ultimately propose, prototype, and estimate recommended results.

C. *Demerits*

1) There are did not exercising artificial intelligence.
2) Security risks.
3) Advanced time complexity for performance process.
4) Limited scalability.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

IV. MODULES

The Module Description for Network trouble Discovery using Artificial Intelligence explaining the functionalities are as follows:

A. Module 1 Data Pre-processing

confirmation ways in machine Learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset.

However, you may not need the confirmation ways, If the data volume is large enough to be representative of the population. still, in real- world scripts, to work with samples of data that may not be a true representative of the population of given dataset. To find the missing value, indistinguishable value and description of data type whether it's pier variable or integer.

The evaluation becomes more prejudiced as skill on the confirmation dataset is incorporated into the model configuration. The confirmation set is used to estimate a given model, but this is for frequent evaluation. It as machine Learning masterminds uses this data to fine- tune the model hyperactive parameters. During the process of data identification, it helps to understand your data and its parcels; this knowledge will help you choose which algorithm to use to make your model.

Many different data drawing tasks use Python's Pandas library and specifically, it concentrates on presumably the biggest data drawing task, missing values and it suitable to more snappily clean data. It wants to spend lower time drawing data, and further time exploring and modeling.

```
df.duplicated()

0         False
1         False
2         False
3         False
4         False
          ...
123112     True
123113     True
123114     True
123115     True
123116     True
Length: 123117, dtype: bool

sum(df.duplicated())
```

Fig 4.1.1 Data pre-processing and cleaning

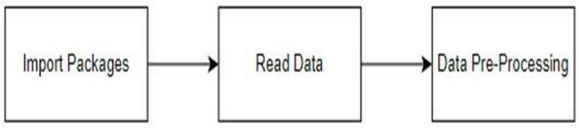


Fig 4.1.2 Module Diagram of Data pre-processing

B. Module 2: Data and Visualization

Data visualization is an important skill. Statistics does indeed concentrate on quantitative descriptions and estimations of data. This can be helpful when exploring and getting to know a dataset and can help with relating patterns, loose data, outliers, and much further. With a little sphere knowledge, data visualizations can be used to express and demonstrate pivotal connections in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves, and it will recommend a deeper dive into some the books mentioned at the end. Sometimes data doesn't make sense until it can look at in a visual form, analogous as with charts and plots. It will discover the multitudinous types of plots that you will need to know when imaging data in Python and how to use them to more understand your own data. Pre-processing refers to the changeovers applied to our data before feeding it to the algorithm. In other words, whenever the data is gathered from different sources it's collected in raw format which isn't realizable for analysis. To achieve better results from the applied model in Machine knowledge system of the data must be in a proper manner. Some specified Machine Learning model needs information in a specified format, for illustration, Random Forest algorithm doesn't support null values. Therefore, to execute an arbitrary timber algorithm null value must

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

be managed from the original raw data set.

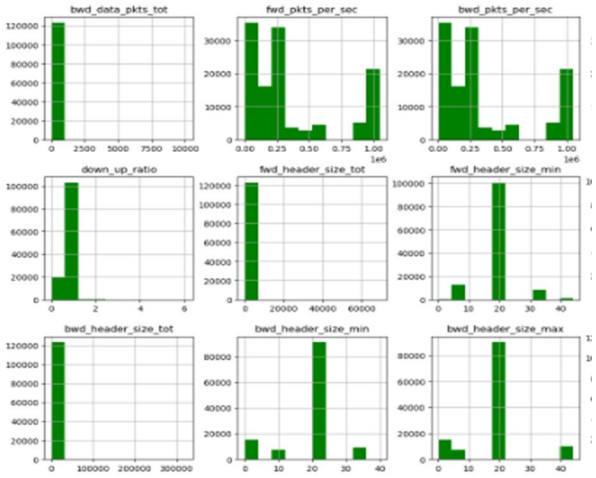


Fig 4.2.1 Graph representation of Input fields

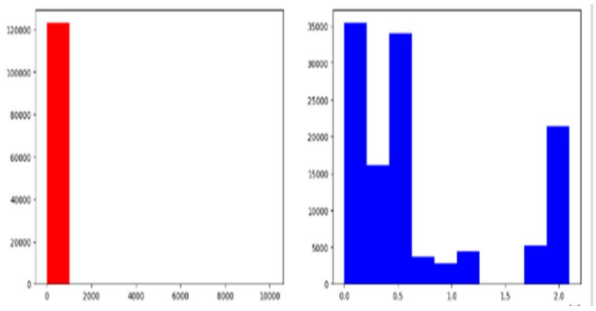


Fig 4.2.2 Bar graph representation of processed data

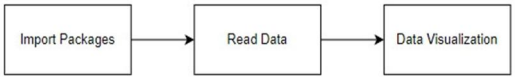


Fig 4.2.3 Module Diagram of Data Analysis and Visualization.

C. Module 3: Random Forest Classifier Algorithm

Random Forest is an ensemble learning algorithm that can be used for both classification and regression tasks.

Random Forest starts by creating multiple subsets of the original dataset through a process called bootstrapping. This involves randomly sampling the data with replacement, creating new datasets of the same size as the original. For each subset, a decision tree is constructed. Decision trees are built by selecting the best feature from a random subset of features at each node, considering various criteria such as Gini impurity for classification or mean squared error for regression. Once all the trees are built, they "vote" for the class (in classification) or provide a prediction (in regression) for a new data point. For regression, the predictions are averaged. One of the key advantages of Random Forest is that it reduces overfitting.

Each tree in the forest is trained on a different subset of the data, and by averaging or voting, the model becomes more robust and less prone to the noise present in individual trees. Random Forest provides a measure of feature importance. Features that are more frequently used for splitting in the trees are considered more important. This information can be valuable for understanding the underlying patterns in the data. Random Forest is a versatile and powerful algorithm that is widely used in practice, especially in situations where interpretability is not the primary concern and high predictive accuracy is desired.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

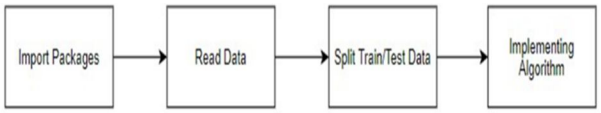


Fig 4.3.1 Module Diagram of Random Forest Classifier

```
# Check the confusion matrix for this algorithms.
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,predicted)
print('THE CONFUSION MATRIX SCORE OF RANDOMFOREST CLASSIFIER:\n\n\n',cm)
```

```
THE CONFUSION MATRIX SCORE OF RANDOMFOREST CLASSIFIER:


[[18453     0     0     0     0     0     0     0   478     0     1     0]
 [   28 18904     0     0     0     0     0     0     0     0     0     0]
 [    0     0 18932     0     0     0     0     0     0     0     0     0]
 [    0     0     0 18932     0     0     0     0     0     0     0     0]
 [    0     0     0     0 18932     0     0     0     0     0     0     0]
 [    0     0     0     0     0 18932     0     0     0     0     0     0]
 [    0     0     0     0     0     0 18932     0     0     0     0     0]
 [    0     0     0     0     0     0     0 18932     0     0     0     0]
 [   90   335     0     0     9     0     0     0 18498     0     0     0]
 [    0     0     0     0     0     0     0     0     0 18931     0     0]
 [  196     0     0     0     0     0     0     0    29     0 18706     0]
 [    0     0     0     0     0     0     0     0     0     0     0 18932]]
```

```
# Check the cross value score of this algorithm.
from sklearn.model_selection import cross_val_score
accuracy = cross_val_score(RFR, x, y, scoring='accuracy')
print('THE CROSS VALIDATION TEST RESULT OF ACCURACY :\n\n\n', accuracy*100)
```

```
THE CROSS VALIDATION TEST RESULT OF ACCURACY :
```

Fig 4.3.2 Random Forest Classifier Accuracy

D. Module 4: AdaBoost Classifier Algorithm

AdaBoost, short for Adaptive Boosting, is an ensemble learning algorithm that is used to boost the performance of weak learners to create a strong classifier. AdaBoost starts with a weak learner, often a simple model like a decision stump. The weak learner's performance is only slightly better than random chances. During each iteration, AdaBoost assigns weights to the training instances. Misclassified instances receive higher weights, focusing the subsequent weak learners on the more challenging examples.

After each iteration, the weights of misclassified instances are increased, directing the algorithm's attention to the previously misclassified samples. The final strong classifier is created by combining the weak learners. Highly accurate weak learners are given more influence. In the testing phase, each weak learner votes on the classification of an instance, and their votes are combined with weights. The final prediction is determined by the weighted majority vote.

```
RFR = AdaBoostClassifier()
RFR.fit(x_train, y_train)
```

```
# Predict is the test function for this algorithm
predicted = RFR.predict(x_test)
```

```
# Check classification report for this algorithm
from sklearn.metrics import classification_report
cr = classification_report(y_test,predicted)
print('THE CLASSIFICATION REPORT OF ADABOOST CLASSIFIER:\n\n',cr)
```

```
# Check the confusion matrix for this algorithms.
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,predicted)
print('THE CONFUSION MATRIX SCORE OF ADABOOST CLASSIFIER:\n\n\n',cm)
```

```
# Check the accuracy score of this algorithms.
from sklearn.metrics import accuracy_score
a = accuracy_score(y_test,predicted)
print("THE ACCURACY SCORE OF ADABOOST CLASSIFIER IS :",a*100)
```

Fig 4.4.1 AdaBoost Classifier Accuracy

E. Module 5: Bernoulli Naive Bayes classifier

The Bernoulli Naive Bayes classifier is a type of probabilistic machine learning model that is particularly well-suited for binary/boolean features. It is based on Bayes' Theorem, which describes the probability of an event based on prior knowledge of conditions that might be related to the event. In the context of the Bernoulli Naive Bayes classifier, the model assumes that each feature follows a Bernoulli distribution, meaning each feature is binary and can take only one of two possible values. This classifier is especially useful for tasks such as text classification.

The Bernoulli Naive Bayes algorithm works by calculating the likelihood of each feature being present given the class label and combines these likelihoods to determine the most probable class for a given instance.

One of the key assumptions of this algorithm is the independence of features, meaning it assumes that the presence or absence of one feature does not affect the presence or absence of any other feature.

This "naive" assumption simplifies the computation and allows the model to scale efficiently to large datasets. Despite this simplification, the Bernoulli Naive Bayes classifier often performs remarkably well in practice, particularly in text-related tasks such as spam detection, sentiment analysis, and document classification.

```
# Split the datasets into two parts like trainng and testing variable.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=42,
print("NUMBER OF TRAIN DATASET    : ", len(x_train))
print("NUMBER OF TEST DATASET     : ", len(x_test))
print("TOTAL NUMBER OF DATASET    : ", len(x_train)+len(x_test))
```

```
NUMBER OF TRAIN DATASET    :  908726
NUMBER OF TEST DATASET     :  227182
TOTAL NUMBER OF DATASET    :  1135908
```

```
print("NUMBER OF TRAIN DATASET    : ", len(y_train))
print("NUMBER OF TEST DATASET     : ", len(y_test))
print("TOTAL NUMBER OF DATASET    : ", len(y_train)+len(y_test))
```

```
NUMBER OF TRAIN DATASET    :  908726
NUMBER OF TEST DATASET     :  227182
TOTAL NUMBER OF DATASET    :  1135908
```

Fig 4.5.1 Bernoulli Naive Bayes classifier Accuracy

```
[[  112    73     0   936  1456   271     0   131  5927    22  9108   896]
 [    0 13228     0  5306     0   224     0     0   174     0     0     0]
 [    0     0 17057     0     0     0     0     0  1875     0     0     0]
 [    0    20     0 18842     0    37     0     8     3     6    16     0]
 [    0   517     0   536 13760     0     0     0  2034     0  2085     0]
 [    0     0     0     0   669 16216     0     0     0     0  2047     0]
 [    0     0     0     0     0     0 18932     0     0     0     0     0]
 [    0    35     0     0     0     0     0 18897     0     0     0     0]
 [   19  1020     0     0    10    23     0     0 17776     0    81     3]
 [    0     0     0     0     9   125     0     0     0 18762    35     0]
 [  359     0     0    40  6324     2     0    21   529     0 10399  1257]
 [   73   221     0  7530  2205   697     0   142  1244     0  4157  2663]]
```

```
# Check the accuracy score of this algorithms.
from sklearn.metrics import accuracy_score
a = accuracy_score(y_test,predicted)
print("THE ACCURACY SCORE OF BERNOULLIINB CLASSIFIER IS :",a*100)
```

```
THE ACCURACY SCORE OF BERNOULLIINB CLASSIFIER IS : 73.35264237483602
```

```
# Check the hamming loss of this algorithm.
from sklearn.metrics import hamming_loss
hl = hamming_loss(y_test,predicted)
print("THE HAMMING LOSS OF BERNOULLIINB CLASSIFIER IS :",hl*100)
```

```
THE HAMMING LOSS OF BERNOULLIINB CLASSIFIER IS : 26.64735625163963
```

Fig 4.5.2 Matrix of Bernoulli Naive Bayes classifier

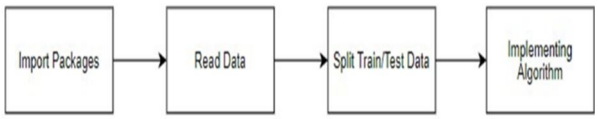


Fig 4.5.3 Module Diagram of Random Forest Algorithm

F. Performance Grounded on Accuracy

Logistic retrogression algorithm also uses a direct equation with independent predictors to prognosticate a value. The prognosticated value can be anywhere between negative perpetuity to positive perpetuity. It needs the affair of the algorithm to be classified variable data.

False Cons (FP) A person who'll pay prognosticated as defaulter. When factual class is no, and prognosticated class is yes. E.g. if the actual class says this passenger didn't survive but the prognosticated class tells you that this passenger will survive.

False Negatives (FN) A person who overpasses prognosticated as payer. When the actual class is yes but the prognosticated class is no. E.g. However, the prognosticated class tells you that passenger will die, if factual class value indicates that this passenger survived.

True Cons (TP) A person who'll does not pay prognosticated as a defaulter. These are the rightly prognosticated positive values which means that the value of factual class is yes, and the value of prognosticated class is also yes.

True Negatives (TN) A person who overpasses prognosticated as payer. These are the rightly prognosticated negative values, which means that the value of an factual class is no, and the value of a prognosticated class is also no. E.g. if a factual class says this passenger didn't survive and the prognosticated class tells you the same thing.

True Positive Rate (TPR) = TP/(TP FN)

False Positive rate (FPR) = FP/(FP TN)

delicacy The Proportion of the total number of prognostications that's correct else overall how frequently the model predicts rightly defaulters and non-defaulters.

delicacy computation

delicacy = (TP TN) (TP TN FP FN)

Delicacy is the most intuitive performance measure, and it's simply a rate of rightly prognosticated observation to the total compliances.

One may suppose that, if we've high delicacy also our model is stylish. Yes, delicacy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are nearly the same.

Precision The proportion of positive prognostications that are correct.

Precision = TP/(TP FP)

Precision is the rate of prognosticated positive compliances to the total prognosticated positive compliances. The question that this metric answer is of all passengers that labelled as survived, how numerous survived? High perfection relates to the low false positive rate. We've got 0.788 perfection which is enough good.

Recall The proportion of positive observed values rightly prognosticated. (The proportion of factual defaulters that the model will rightly prognosticate)

Recall = TP/(TP FN)

Recall(perceptivity)- Recall is the rate of rightly prognosticated positive compliances to the all compliances in factual class- year.

F1 score takes both false cons and false negatives into account. Intimately it isn’t as easy to understand as delicacy, but F1 is generally more useful than delicacy, especially if you have an uneven class distribution. Delicacy works best if false cons and false negatives have analogous cost.

General Formula

F- Measure = 2TP/ (2TP FP FN)

F1- Score Formula

F1 Score = 2 *(Recall * Precision) (Recall Precision)

Module 6: Deployment

The system integrates machine learning models into a Django-based web application with an intuitive user interface, enabling real-time intrusion detection. This seamless integration allows users to monitor and respond to risks effectively within networks. Predicting the output whether the given image is CKD / Not CK.

Fig 4.6.1 User Interface

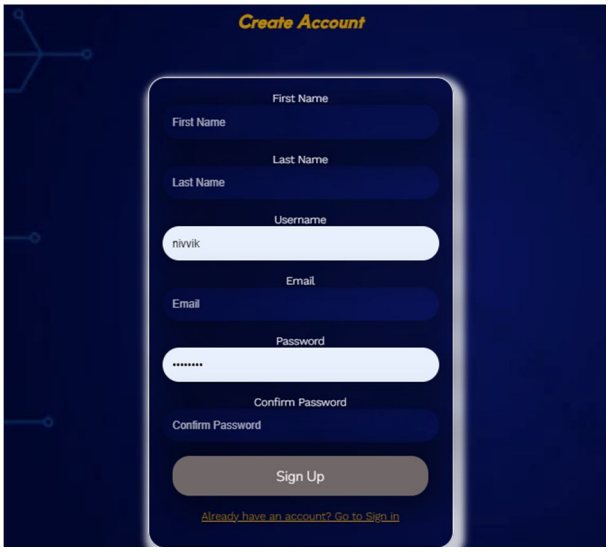


Fig 4.6.2 Register Page

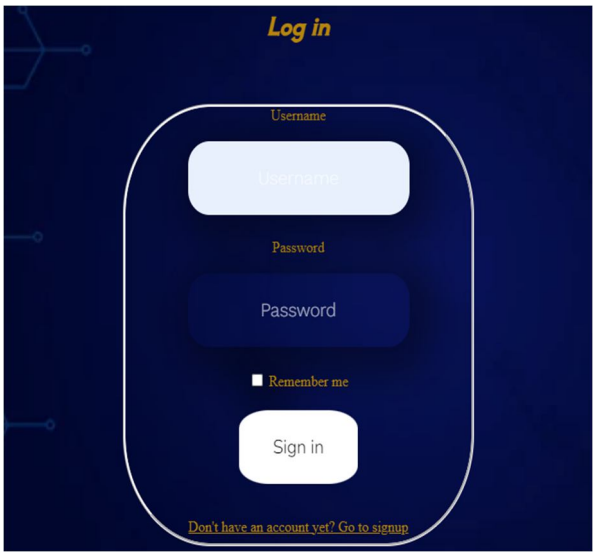


Fig 4.6.3 Login Page

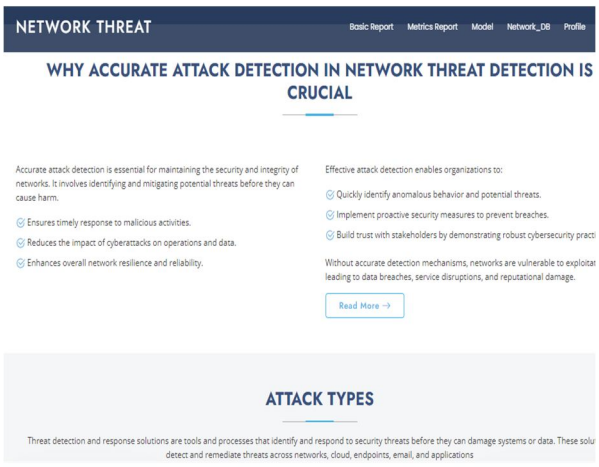


Fig 4.6.4 Home Page

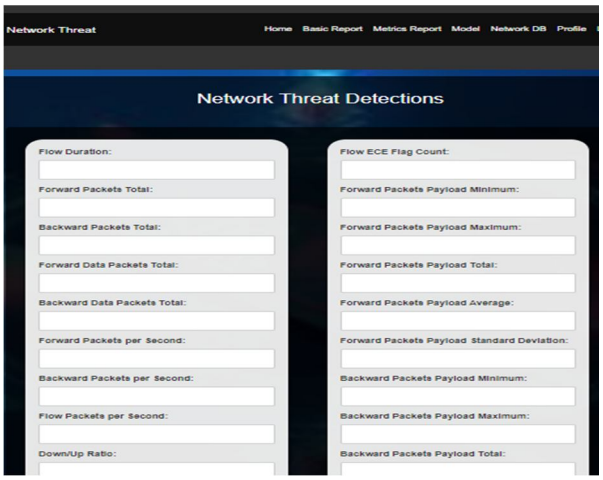


Fig 4.6.5 Input Page

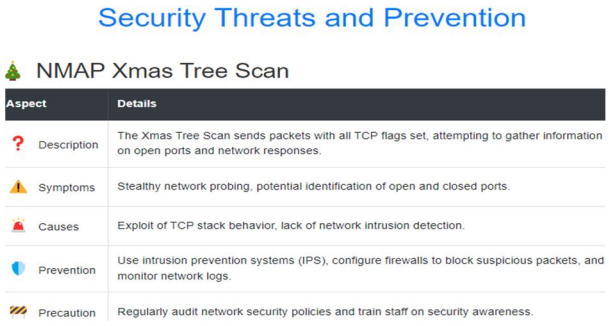


Fig 4.6.6 Output Page

V. RESULTS AND DISCUSSION

The proposed system's performance was evaluated through various tests, with results showing a substantial improvement over the existing system. The existing method, which uses side-channel power consumption analysis, demonstrated limited detection capabilities, especially for covert attacks that do not cause significant power spikes. It also struggled with scalability in dynamic IoT environments, with accuracy decreasing as network complexity increased. In comparison, the proposed system, which integrates Random Forest, Bagging, and Ridge Classifiers within a Django-based framework, significantly improved detection accuracy.

The machine learning classifiers achieved high precision and recall, with Random Forest scoring 99% in both. This method not only handled diverse attack types but also provided real-time monitoring through an interactive Django interface, making it a scalable and flexible solution for various IoT environments.

The system's efficiency is further demonstrated by its ability to classify multiple attacks in real-time, providing immediate results to users via the web interface. Additionally, it showed improved scalability and adaptability, accommodating varying IoT network sizes and dynamic attack patterns. By leveraging machine learning's predictive capabilities, the proposed system ensures better detection rates and lower false positives, offering a more reliable solution for securing IoT networks.

Overall, the proposed system provides a robust, scalable, and user-friendly solution for IoT security, addressing the limitations of the current power-based detection methods and offering enhanced real-time threat analysis.

A. Comparison of Existing and Proposed System

Current cellular IoT infrastructures focus on processing large-scale IoT services with the help of LTE-M and NB-IoT technologies. Security is compromised by system-immanent weaknesses and implementation errors. The infrastructures are reactive and detect weaknesses only after the occurrence of attacks, and they do not have strong prediction capabilities. By leveraging artificial intelligence-powered anomaly detection and sophisticated deep learning methods, it continuously monitors network traffic to identify and neutralize threats before they can be exploited.

It goes beyond traditional IoT threat management by offering real-time detection and dynamic learning capabilities. Use of IoT services over common cellular infrastructures typically brings security exposures because of inadequate implementation practices, typical design errors, and variable configurations. Threats like remote IP exposure, power-saving service disruption, and SMS/data spamming occur because of uncoordinated integration. The proposed model is designed with integration in mind, so predictive modelling becomes an organic extension of the network security architecture. It allows monitoring at both system and device levels, thus reducing the attack surfaces due to integration mistakes. Attack evaluations show that attackers can exploit cellular IoT vulnerabilities quickly and economically, such as draining battery life, spamming devices, or ballooning data bills within minutes. Such systems lack automated defines mechanisms, and hence the response is delayed and the potential harm is greater. One of the key benefits of the suggested system is that it can detect and react in real time. By using artificial intelligence to continuously scan for anomalies, the system can recognize suspicious activity in real time and trigger alerts or corrective actions—thus reducing damage and lessening dependence on human action during critical response times.

Legacy systems are static and not adaptive. Risks discovered will have to be patched manually and updated periodically using routine updates that may not get rolled out consistently and in time on all equipment and networks. The proposed system's AI models are trained in huge volumes of past threats and are capable of learning and evolving over time. With each new data point and occurrence, the system becomes smarter, recognizing even previously unknown attack patterns. This adaptive learning guarantees long term security and greater efficiency. Though adequate for general service provision, existing cellular IoT deployments have serious security vulnerabilities. Inadequate predictive modelling, real-time inspection, and built-in defenses make them susceptible to advanced, low-cost attacks. The proposed AI-empowered solution dramatically enhances network cybersecurity posture. It offers a solution that, in terms by predictive analytics, swift incident handling, and continuous learning, features that are fundamentally essential in guaranteeing the integrity of critical infrastructure in the face of contemporary threat.

While existing cellular IoT infrastructures provide the foundation for deploying massive IoT, they do not have proactive threat management. Our proposed AI-fortified predictive modelling system addresses these shortcomings by offering smart, real-time, and adaptive defense, establishing a future-proofed cybersecurity infrastructure fit for the future digital environment.

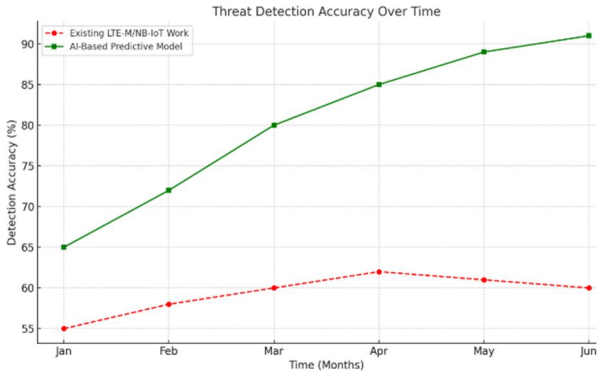


Figure 5.1.1 Comparison of the proposed system with the existing system

1) The AI-based model improves steadily from 65% to over 90% in six months.
2) The traditional LTE-M/NB-IoT method shows only slight improvement and then dips.
3) The performance gap between AI and the traditional method keeps getting wider.
4) AI continues to learn and improve, while the traditional method levels off.
5) AI provides consistent results, whereas the traditional method is more unstable.

VI. CONCLUSION

Our network threat detection predictive modelling strategy is based on the principles of sophisticated artificial intelligence methods. Utilizing machine learning algorithms that can scan large datasets, the system has the capability to detect patterns, anomalies, and unusual behaviors that could be potential signs of cyber threats. Through this data-driven approach, proactive cybersecurity is possible, going beyond the conventional reactive measures that heavily depend on known threat signatures and human intervention. The use of AI facilitates real-time processing of data and threat detection, which is essential in the current rapid-paced digital landscape. Since threats can arise and propagate within a matter of seconds, responding quickly while being able to detect them is essential.

Our platform continuously scans network traffic, raising red flags on suspicious activities and initiating proper alerts or responses automatically. This essentially shortens response time, limiting possible losses and providing organizations with a vital advantage in counteracting attacks.

Another key benefit of our AI-based system is its ongoing flexibility. In contrast to rigid rule-based systems, our model improves by learning from new information and previous events, becoming more accurate with time. This flexibility makes the system continue to be effective even as cyber threats become increasingly sophisticated and advanced. It is also able to adapt to various network environments and threat profiles, thus being a scalable and versatile solution for numerous use cases. Finally, this smart threat detection system provides a strong and resilient method for ensuring network integrity.

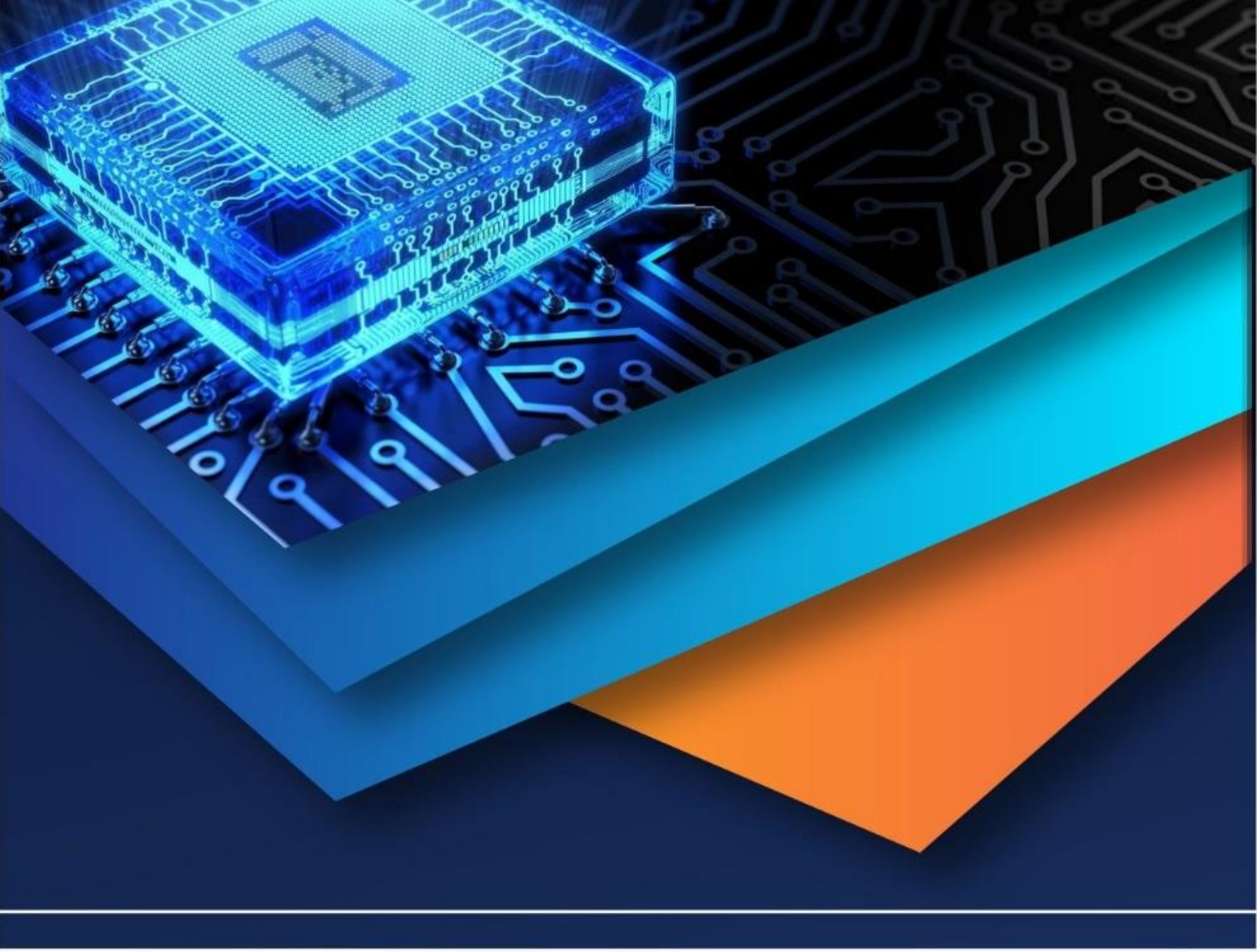
By improving detection accuracy, speeding up response times, and learning from the constantly evolving cyber environment, the system offers a reliable definition against unauthorized access, data breaches, and other security threats. Not only does it protect sensitive data, but it also facilitates long-term cybersecurity strategies for organizations looking to remain ahead of cyber attackers.

REFERENCES

[1] Abusitta, A., Li, M. Q., & Fung, B. C. M. (2021). Malware classification and composition analysis: A survey of recent developments. Journal of Information Security and Applications,59,102828. https://doi.org/10.1016/j.jisa.2021.102828
[2] R. Bharathi, R. Bhavani, and R. Priya, Twitter text sentiment analysis of Amazon unlocked mobile reviews using supervised learning techniques, Indian J. Compute. Sci. Eng., vol. 13, no. 4, pp. 1242-1251, 2022. [Online]. Available: https://www.ijcse.com/docs/INDJCS E22-13-04-100.pdf
[3] Habibi,O.,Chemmakha,M.–& Lazaar, M. Performance Evaluation of CNN and Pre-trained Models for Malware Classification. Arab J Sci Eng **48**,10355–10369,(2023). https://doi.org/10.1007/s13369-023-07608-z
[4] R. Bharathi, R. Bhavani, and R. Priya, Leveraging deep learning with sentiment analysis for Online Book reviews polarity classification model, Multimed. Tools Appl., 2024. Available: https://doi.org/10.1007/s11042-024-20369-7.
[5] Pachpute, S. S. (2019). Malware Analysis on PDF. Master's Projects. San Jose State University. https://doi.org/10.31979/etd.pf8d-htjh.
[6] R. Bharathi, "Study of Comparison between Bat Algorithm, Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO) for user's bank loan and their related due history," International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), vol. 3, issue,5,pp.1168-1176,May-June,2018. https://ijsrcseit.com/home/issue/view/article.ph p?id=CSEIT1835264
[7] Atadoga, A., Sodiya, E. O., Umoga, U. J., & Amoo, O. O. (2024). A comprehensive review of machine learning's role in enhancing network security and threat detection. World Journal of Advanced Research and Reviews, 21(02), 877–886. https://doi.org/10.30574/wjarr.2024.21.2.0501
[8] Nguyen, T. T., & Reddi, V. J. (2021). Deep Reinforcement Learning for Cyber Security. IEEE Transactions on Neural Networks and Learning Systems,32(11),5239–5253. https://doi.org/10.1109/TNNLS.2021.3121870
[9] Okoli, U. I., Obi, O. C., Adewusi, A. O., & Abrahams, T. O. (2024). Machine learning in cybersecurity: A review of threat detection and defense mechanisms. World Journal of Advanced Research and Reviews, 21(01), 2286–2295. https://doi.org/10.30574/wjarr.2024.21.1.0315
[10] R. Bharathi, R. Bhavani, & R. Priya. "Leveraging Deep Learning with Sentiment Analysis for Online Book Reviews Polarity Classification Model", Multimedia Tools and Applications, 17 October 2024, pp,1-20.DOI:https://doi.org/10.1007/s11042-024-20369-7
[11] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A Deep Learning Approach to Network Intrusion Detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41-50. https://doi.org/10.1109/TETCI.2017.2772792
[12] Bharadiya J. Machine learning in cybersecurity: Techniques and challenges. Eur J Technol. 2023;7(2):1–14.
[13] Dasgupta D, Akhtar Z, Sen S. Machine learning in cybersecurity: A comprehensive survey. J Def Model Simul. 2022;19(1):57–106.
[14] National Institute of Standards and Technology (NIST). Framework for improving critical infrastructure cybersecurity. [Internet]. Available from: https://www.nist.gov/publications/frameworkimproving-critical-infrastructurecybersecurity-version-11
[15] IBM Security. AI cybersecurity. IBM Study; 2024. Available from: https://www.ibm.com/security/artificialintelligence
[16] Microsoft. Microsoft cloud security for enterprise architects. [Internet]. Available from: https://download.microsoft.com/download/6/d/f/6dfd7614-bbcf-4572-a871-e446b8cf5d79/msft_cloud_architecture_security.pdf
[17] Cisco Security. Annual cybersecurity report. [Internet]. Available from: https://engage2demand.cisco.com/LP=9810
[18] Google Cloud. Security & identity. Google Cloud; 2024. [Internet]. Available from: https://cloud.google.com/blog/products/identity-security
[19] Amazon Web Services (AWS). AWS cloud security. [Internet]. Available from: https://aws.amazon.com/security
[20] CloudFlare. Trends report: State of application security in 2024. [Internet]. Available from: https://www.cloudflare.com/en-in/2024-application-security-trends
[21] Symantec. Enterprise security. Broadcom Software; 2024. [Internet]. Available from: https://www.broadcom.com/solutions/enterprise-security
[22] Altulaihan EA, Alismail A, Frikha M. A survey on web application penetration testing. Electronics. 2023;12:1229.
[23] Sadqi Y, Maleh Y. A systematic review and taxonomy of web applications threats. Inf Secur J Glob Perspect. 2022;31:1–27.
[24] Otuu, Obinna Ogbonnia, and Felix ChukwumaAguboshim. "Aguide to the methodology and system analysis section of a computer science project." World Journal of Advanced Research and Reviews 19.2 (2023): 322-339
[25] Spring, Jonathan M., et al. "Machine learning in cybersecurity: A Guide." SEI- CMU Technical Report 5 (2019).
[26] Alhamed M, Rahman MMH. A systematic literature review on penetration testing in networks: Future study directions. Appl Sci. 2023;13:6986.
[27] Sarker, Iqbal H. "Machine learning for intelligent data analysis and automation in cybersecurity: current and future prospects." Annals of Data Science 10.6 (2023): 1473-1498.
[28] Makino Y, Klyuev V. Evaluation of web vulnerability scanners. In: 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems; 2015 Sep 24–26; Warsaw, Poland. Vol. 1. p. 399–402.
[29] Liu, Jing, Yang Xiao, Shuhui Li, Wei Liang, and C. L. Philip Chen. "Cyber Security and Privacy Issues in Smart Grids." IEEE Communications Surveys and Tutorials/IEEE Communications Surveys and Tutorials 14, no. 4 (January 1, 2012): 981–97. https://doi.org/10.1109/surv.2011.122111.00145.

[30] F Kagorora, Li J, D Hanyurwimfura, L Camara Effectiveness of web application security scanners at detecting vulnerabilities behind AJAX/JSON. Int J Innov Res Sci Eng Technol. 2015; 4:4179–88.
[31] Singh N, Meherhomji V, Chandavarkar BR. Automated versus manual approach of web application penetration testing. In: 2020 11th Int Conf on Computing, Communication and Networking Technologies (ICCCNT); 2020 Jul 1–3; Kharagpur, India. p. 1–6.
[32] Hu Z, Beuran R, Tan Y. Automated penetration testing using deep reinforcement learning. In: 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW); 2020 Sep 7–11; Genoa, Italy. p. 2–10.
[33] Hance J, Milbrath J, Ross N, Straub J. Distributed attack deployment capability for modern automated penetration testing. Computers. 2022;11:33.
[34] Elmrabit N, Zhou F, Li F, Zhou H. Evaluation of machine learning algorithms for anomaly detection. In: 2020 Int Conf on Cyber Security and Protection of Digital Services; 2020 Jun 15–19; Dublin, Ireland. p. 1–8.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)