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.2025.69739

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

Enhancing IoT Security using Feature Reduction Techniques

Dr.Pradeep Kumar[1], Prakhar Gupta[2], Harsh Som[3], Om Gupta[4]
Department of Computer Science and Engineering, J.S.S Academy of Technical Education, Noida, India

Abstract: As Internet of Things (IoT) devices become more common, they're also becoming a biggertarget for cyberattacks. Thesedevicesoften have limited resources, making it tough to keep them secure. One promising solution is using machine learning in Network Intrusion Detection Systems (NIDS). However, for these systems to work efficiently, they need to reduce the number of features they analyze—this helps save computational power without sacrificing accuracy. In this study, we take a close look at different ways to reduce features, comparing techniques like feature selection (FS)—such as Pearson Correlation and Chi-square—with feature extraction (FE) methods like Principal Component Analysis (PCA) and Autoencoders (AE). We tested these approaches using two IoT- specific datasets, TON-IoT and BoT-IoT, and evaluated five machine learning models: Decision Tree,RandomForest,k-NearestNeighbors,Naive Bayes, and Multi-Layer Perceptron. The results show that FE methods tend to perform better in terms of accuracy and robustness, especially with complex datasets, though they require more computational power. On the other hand, FS techniques like Chi-square offer a good balance between performance and efficiency. Among FE methods,PCAisfasterthanAE.Interestingly,FS works better with smaller datasets, while FE is more effective for handling a variety of attack types.Thisresearchprovidespracticaladviceon choosing the right feature reduction method for IoT environments, helping strike a balance between accuracy and computational efficiency. These insights are crucial for building scalable, real-time NIDS that can handle the unique challenges of IoT systems, benefiting both researchers and industry professionals working on IoT security.

Index Terms: Internet of Things (IoT), Feature Extraction (FE), Principal Component Analysis (PCA), Autoencoders (AE), Network Intrusion Detection Systems (NIDS).

I. INTRODUCTION

The Internet of Things (IoT) has revolutionized how we live and work, connecting billions of devices to collect,share,andanalyzedata.Fromsmarthomesto healthcare and industrial automation, IoT is everywhere. By 2025, experts predict there will be nearly 30.9 billion IoT devices in use, highlighting their growing importance. But great connectivity is also accompanied by great risk—IoT devices are usually easy targets for cyber threats such as DDoS, ransomware, and data breaches. Their limited computing resources and usually weak security controls leave them particularly exposed, an exigent demand for good, IoT-specific security solutions.

Welcome Network Intrusion Detection Systems (NIDS)basedonmachinelearning.Theyareaclever method to discover and block cyber threats. But IoT networksproduceenormousamountsofdata,andthe devices themselves are resource-constrained, so traditional approaches to security are less effective. That is where feature reduction methods step in. By reducing the complexity of the data—either by choosing the most significant features (Feature Selection,orFS)orconvertingitintoamorereduced form (Feature Extraction, or FE)—we can improve the efficiency of NIDS without losing key information. FS techniques such as Pearson Correlation and Chi-square assist in identifying the most salient features, whereas FE methods such as Principal Component Analysis (PCA) and Autoencoders (AE) transform the data into a more compact form.

This project focuses on improving IoT security by testing and comparing these FS and FE methods on two IoT-specific datasets, TON-IoT and BoT-IoT. Thegoalistofindthebestbalancebetweenaccurate threat detection and efficient use of resources, ensuringNIDScankeepupwiththeuniquedemands of IoT environments.

II. METHODOLOGY

We present the following section where feature reduction techniques are applied to a pipeline of machine learning-based NIDS. It aims to experimentally evaluate how such techniques may influence the classifiers' performance. Figure 1 schematically outlines the scheme including data preprocessing,classbalancing,featurereduction,and classification. Below, we present the three phases in which we had divided the proposed model, with special emphasis on the two key feature reduction methods.

1) Phase1:DataPreprocessingandClass Balancing

Beforeperformingtheanalysis,therawdatafromthe TON-IoT and BoT-IoT datasets was prepared for processing. Data Cleaning: This involved handling missingvalues,removingduplicateentries,encoding non-numericalfeatureslikecategories,andstripping out identifying information that may skew results. Data Splitting: The cleaned data then splits into a training set and a testing set for proper, unbiased evaluation. Normalization:Features are scaled intoa standard range; then the features are uniform throughout the data to ensure that none of them dominatetheothers.ClassBalancing:Tocounterthe imbalanced classes, where certain attack types are underrepresented, the Synthetic Minority Over- samplingTechnique(SMOTE)isemployed.SMOTE generates synthetic samples for minority classes to prevent the model from being biased towards the majorityclass.Thisstepensuresthatthedataisclean, balanced, and ready for analysis.

2) Phase 2: Dimensionality Reduction Evaluation

Thisphase,wesimplifythedatafortheimprovement of the performance of the models and reduce the computational overhead. Two main approaches, based on the theme of dimensionality reduction, are implemented:FeatureSelection:Thismethodsimply includes the most important features about the analysis. Techniques include:Pearson Correlation; measures the linear relationship between features; Average Correlation Score; summarizes the importance of the feature; Threshold Schemes, simply select features based on pre-defined criteria; Chi-Square Test, ranking based on the statistical significance. Feature Extraction: Thattransformsthe data into a lower-dimensional space. Techniques include:PrincipalComponentAnalysis,basedonthe theme of maximizing variance; Autoencoders, based onthethemeofminimizingreconstructionerror.This themesimplyaimstocapturenon-linearrelationships betweenthefeatures;bothsimplyensurethecompact representationofthedata;therefore,easiertoprocess while retaining critical information.

3) Phase3:Classification

Inthefinalphase,thereduceddatasetisusedtotrain and test various machine learning classifiers, including:

- DecisionTrees(DT)
- RandomForests(RF)
- k-NearestNeighbors(kNN)
- NaiveBayes (NB)
- Multi-LayerPerceptron(MLP)

Theprocess involves:

Model Training: Fitting theclassifiers to the training data.

Model Testing: Evaluating their performance on the test data.

Performanceismeasuredusingmetricslikeaccuracy, precision, recall, F1-score, feature reduction time, model training time, and prediction time. This phase helps us understand how well each feature reduction technique improves the classifiers' ability to detect and classify attacks accurately and efficiently.

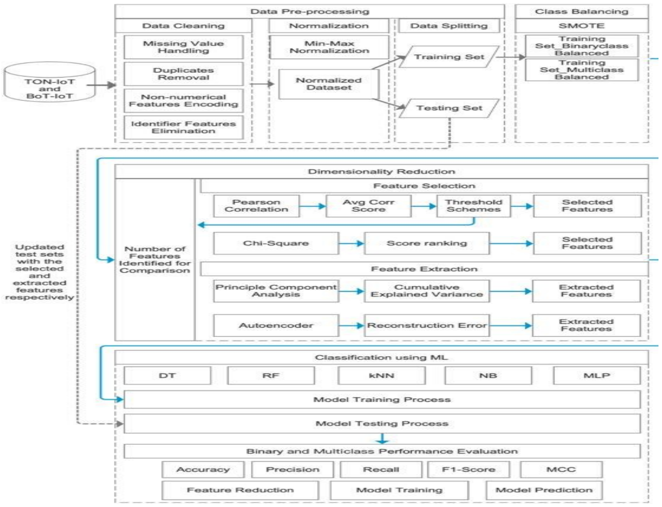


Fig. 1.The proposed NIDS framework evaluates feature reduction methods to enhance IoT security

III. DATASETS

InordertotrainandassessIoT-orientedclassification models, this research employs two openly published datasets with high IoT orientation: TON-IoT (Moustafa, 2021) and BoT-IoT (Koroniotis et al., 2019). In the following, we describe these datasets and their distinguishing features.

A. TON-IoT

The TON-IoT dataset is an advanced, exhaustive resource that aims to challenge AI-based security systems in IoT networks. It was developed based on a distributed architecture that integrates edge, fog, and cloudcomputinglayers,usingtechnologiessuch as Software-Defined Networking (SDN), Network Function Virtualization (NFV), and Service Orchestration (SO). This configuration enables the development of dynamic, realistic testbed networks that simulate real-world IoT scenarios.

The dataset contains heterogeneous data sources, including:

- IoTservicetelemetry data.
- WindowsandLinuxoperatingsystem logs.
- Dataonnetworktraffic.

TON-IoT is unique because it records both benign and attack activity in IoT networks and thus forms a solid basis for the verification of cybersecurity measures. In contrast to most other datasets, TON- IoTprovidesaricherandmorerealisticrangeofdata, rendering it suitable for the design and testing of AI modelsspecifictosecurecomplexIoTenvironments.

Testbed Overview:

The TON-IoT testbed emulates IoT and edge networks within a smart city setting, organized in three layers:

EdgeLayer:ControlsIoT/IIoTdevicesandgateways. Fog Layer: Processes computing and analytics with virtualization technologies.

Cloud Layer: Incorporates cloud services for telemetry data management and security event simulation.

This multi-layer design guarantees the dataset accurately represents real-world IoT situations.

B. BoT-IoT

The BoT-IoT dataset was created to address the increasing need for realistic, extensive data in network forensic analytics, especially for IoT networks.ItemulatesavarietyofIoTactivities,both normal and attacks, such as Distributed Denial of Service (DDoS), data stealing, and keylogging.

The BoT-IoT testbed was specifically designed to have a combination of valid and malicious behavior, employing a range of IoT devices and tools. This guarantees the dataset reflects the diversity and complexity of real-world IoT scenarios.

TheprincipalreasonsforselectingBoT-IoTare:

1) Itsproperlydocumentedtestbedinstallation. Itsincorporationofrecent,advancedattackpatterns. Robust labeling of data for proper analysis.
2) These characteristics make BoT-IoT a priceless toolfortrainingandtestingmachinelearningmodels designed to identify and counter IoT-specific cyber attacks.

Preprocessing and class balancing are crucial processes involved in preparing the IoT datasets for the design of robust and accurate machine learning- drivenNetworkIntrusionDetectionSystems(NIDS). Theseprocessesensurethatthedataisclean,uniform, and prepared for training dependable models. These also address the prevalent issue of class imbalance, which can prevent the detection of infrequent but essential attack types.

a) Data Preprocessing

Data preprocessing is the process of converting raw IoT data into a format that can be used in machine learning. For this project, TON-IoT and BoT-IoT datasets were utilized, and the following steps were performed to clean and prepare the data:

- *HandlingMissingValues:*

MissingorincorrectvaluesinIoTdatasetscancause noise and decrease model performance.

In the TON-IoT dataset, missing values (denoted as "-") in features such as DNS, SSL, and HTTP were substituted with placeholders ("n/a") to preserve dataset integrity.

For the BoT-IoT dataset, erroneous records, e.g., ARP instances containing "-1" in source or destination ports, were eliminated since they did not add to meaningful network activity.

- *RemovingDuplicates:*

Duplicaterecordscan overinflatethedataset without providingusefulinformationandcanskewthemodel.
Morethan11,000duplicaterowswereremovedinthe TON-IoT dataset to leave only the unique instances. There were no duplicates in the BoT-IoT dataset, so the step was bypassed.

- *EncodingFeatures:*

Bothdatasets'non-numericalfeaturesweretranslated into numerical representations to ensure that they were machine learning model-friendly.
One-Hot Encoding was used for categorical features with unique values, like protocol types (proto) and connection states (conn_state). This converted each categorytoabinaryvector,whichmadeitsimplerfor the model to understand.
Binary Encoding was used for binary outcome features,likeDNSflags(dns_AA,dns_RA)andSSL indicators (ssl_established), making their representation simpler.

- *RemovingIrrelevantFeatures:*

Attributes like timestamps (ts), IP addresses (src_ip, dst_ip), and port numbers (src_port, dst_port) were excluded. These attributes are not informative for classification and can contribute to overfitting. They tendtobeenvironment-specificanddonotgeneralize well to new network conditions.
NormalizingData:
Data normalization makes all features have an equal contribution during training to the model.
Min-Max normalization was applied to normalize featuresintoaconsistentrange(0to1)toavoidbiases due to features with higher values (e.g., packet lengths or packet durations).
This is particularly crucial for algorithms such as k- Nearest Neighbors and Multi-Layer Perceptrons, which are feature magnitude-sensitive.

- *SplittingData:*

For fair evaluation, the datasets were divided into training(80%)andtesting(20%)subsetsbystratified sampling. This maintained class distribution in both subsetssothemodel couldgeneralize well to unseen data.

b) Class Balancing

IoT datasets tend to be class-imbalanced, with some types of attacks (minority classes) being underrepresented in relation to normal traffic (majority class). This class imbalance can lead to machine learning models preferring the majority class, thereby decreasing their performance in detecting infrequent but significant attack types.
Applying SMOTE (Synthetic Minority Over- sampling Technique):
SMOTE was employed to balance the datasets by generatingsyntheticsamplesfortheminorityclasses.

- For Binary Classification: SMOTE balanced the data by creating more samples for the attack class to make it proportional to the normal class.
- For Multi-Class Classification: SMOTE produced synthetic instances for every minority attack class, includingDDoS,DoS,Ransomware,andXSS,sothat all classes were equally balanced.

➢ PreventingOverfitting:

SMOTE was run only onthe trainingset sothat data leakintothetestingsetwasavoided.Thismaintained the integrity of the testing procedure while ensuring that the model was tested on distributions seen in practice.

➢ BalancingSpecificClasses:

TON-IoT and BoT-IoT datasets comprise several attack types, with certain types having heavy distributions. SMOTE targeted minority categories selectively in order to balance the classes so that the model could learn the varied attack patterns better.

IV. FEATURE REDUCTION

Featurereductionisanimportantstepinconstructing effective and scalable machine learning-based NetworkIntrusion DetectionSystems(NIDS)forIoT networks. IoT datasets like TON-IoT and BoT-IoT tend to have high-dimensional data, which can be computationally intensive and impede the performance of machine learning models. The objective of feature reduction is to minimize the number of features but retain the most significant ones, enhancing computational efficiency as well as detection accuracy. This process is split into two primary approaches: Feature Selection (FS) and Feature Extraction (FE).

V. FEATURE SELECTION(FS)

Feature selection is aimed at selecting and retaining the most useful features of the original data without altering its format. It assists in reducing redundancy andmakessurethatonlythemostimportantfeatures areappliedforclassificationpurposes.Thefollowing feature selection methods were employed in this project:

1) Pearson Correlation:

It computes the linear correlation between the features and target variable.

Featuresthathaveastrongcorrelationwiththetarget andaweakcorrelationamongthemselvesarechosen to prevent multicollinearity.

Aniterativeadjustmentofacorrelationcutoffisused to obtain the optimal set of features.

2) Chi-SquareTest:

This statistical test tests the independence of categorical features with the target variable.

Features with high Chi-square measures, which confirm the presence of strong relationships with the target, are selected.

This method is light and efficient in IoT scenarios where there are limited resources.

VI. FEATURE EXTRACTION(FE)

Feature extraction is all about simplifying data. It takes alarge, complexdataset andshrinksit downto a smaller, more manageable size while keeping the most important information intact. This is especially helpful when dealing with IoT data, which can be messy and full of complicated relationships. In this project,weusedtwomaintechniquestoachievethis:

1) PrincipalComponentAnalysis(PCA):

PCA is like a data compressor. It finds the most important patterns in the data and focuses on those, ignoring the less important details. This makes it fasterandeasiertoworkwith,whichisperfectforIoT systemsthatdon’thavealotofcomputingpower.The best part? You can decide how much of the original data to keep, so you don’t lose anything critical.

2) Autoencoders(AE):

Autoencoders are a bit fancier. They’re a type of neuralnetworkthatlearnshowtosqueezedataintoa smaller form and then rebuild it back to its original shape.Thinkofitliketakingahigh-resolutionphoto, saving it as a smaller file, and then zooming back in without losing too much detail. While autoencoders aregreatathandlingcomplexdata,theydoneedmore computing power compared to PCA.

3) AttackClassification

One of the biggest challenges in IoT security is spottingattacksbeforetheycauseharm.That’swhere attack classification comes in. It’s like a security guardthatwatchesoverthenetwork,sortingthrough all the traffic to find anything suspicious. IoT networks are especially vulnerable to attacks like DDoS, ransomware, and injection attacks, so having a system that can accurately identify these threats is crucial. In this project, we used machine learning to build a system that can detect attacks quickly and efficiently, even with limited resources.

VII.CLASSIFICATION GOALS

Our attack classification system has three mainjobs: Tell normal from malicious: It needs to figure out what’s regular traffic and what’s an attack.

Identify specific attacks: It should be able to recognizedifferenttypesofattacks,likeDDoS,DoS, password attacks, and reconnaissance.

Catchtheareones: Evenifanattackisuncommonor doesn’t show up often in the data, the system should still be able to spot it.

Machine Learning Models We Used

To make this system work, we tested five popular machine learning models. Each has its strengths and weaknesses, so we picked the ones that fit best with our goals:

A. DecisionTree(DT):

Decision Treesarelike flowcharts. Theyask aseries ofyes/noquestionstoclassifydata.They'rerelatively simple, easy to understand, and don't require that much computing power, so they're great for very small datasets.

B. RF:

RandomForesttakesDecisionTreestothenextlevel. Insteadofusingjustonetree,itusesawholeforestof them. This makes it more accurate and better at handling complex data, but it’s also a bit heavier on resources.

C. k-NearestNeighbors(kNN):

kNNislikeaneighborhoodwatch.Itlooksatthedata points closest to the one it’s trying to classify and makesadecisionbasedonwhatitsees.Itworkswell when the data is neatly separated, but it can slow down with larger datasets.

D. NaiveBayes(NB):

NaiveBayesisquickandefficient.Itusesprobability to make predictions, assuming that all thefeaturesin thedataareindependent.Thismakesit agoodfit for IoT systems, where resources are often limited.

E. Multi-LayerPerceptron(MLP):

MLPisatypeofneuralnetworkthatcanhandlereally complex data. It’s great for finding patterns in large datasets, but it needs a lot of computing power to do its job.

VIII. CONCLUSION

This 'Improving IoT Security Using Feature Reduction Methods' project aims to mitigate increasing vulnerabilities in IoT networks. IoT networks are often cyberattacked due to their restricted processing capabilities and lack of resources.

Thestudyexploredfeatureselection(FS)techniques, such as Pearson Correlation and Chi-square, with feature extraction (FE) techniques, such as Principal ComponentAnalysis(PCA)andAutoencoders(AE), on actual IoT datasets like TON-IoT and BoT-IoT. The results indicated that FS methods are computationallyless intensive and adequate for real- time applications, while FE methods are more accurateandrobust but comeat thecost of increased computation.

Different machine learning techniques—random trees,randomforests,k-nearestneighbors,andNaive Bayes, Multi-Layer Perceptrons, and Decision Trees—were compared to determine the most effective combinations of feature reduction algorithms and classifiers. The findings illustrated that Autoencoders combined with Random Forests delivered the greatest accuracy, and Chi-square in combination with PCA offered a good blend of computationaleffectivenessandbalanceindetection.

By solving problems like high-dimensional data, classimbalance,andresourcelimitation,thisresearch helps in the creation of scalable, real-time security solutions specific to IoT networks. The findings of this research provide useful insights for both academic researchers and industry professionals looking to design more robust and adaptive security frameworks for IoT ecosystems.

In conclusion, this project highlights the central role of feature reduction methods in future-proofing IoT security. It establishes the foundation for the next generation of innovations in intrusion detection systems that are not only fast and efficient but specially crafted to suit the specific needs of IoT environments.

IX. FUTURE SCOPE

This project lays the groundwork for strengthening IoT security by leveraging feature reduction techniques in Network Intrusion Detection Systems (NIDS). However, there are several avenues for further advancement to enhance IoT security:

ExplorationofAdvancedMachineLearningModels: Future research could investigate the use of sophisticated algorithms such as Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) to boost detection accuracy, especially for intricate and evolving attack patterns.

Real-Time Threat Detection: Efforts can be directed toward developing real-time intrusion detection systems capable of operating with minimal delay, ensuring timely responses in fast-paced IoT environments.

UtilizationofDiverseDatasets:Expandingtherange of datasets to include a wider variety of IoT devices and attack scenarios will help improve the adaptability and robustness of detection models against emerging threats.

Integration with Complementary Security Measures: Combining NIDS with other security tools, such as firewallsandanomalydetectionsystems,cancreatea comprehensive, multi-layered defense strategy for IoT networks.

Development of Lightweight Solutions: Designing energy-efficient and resource-optimized intrusion detection systems is essential to ensure security withoutcompromisingtheperformanceoflow-power IoT devices.

Adversarial Machine Learning Research: Investigating adversarial machine learning techniques can strengthen NIDS against potential threats like data manipulation or evasion attacks, ensuring greater system resilience.

By pursuing these directions, future work can build on this project's foundation to create more effective, scalable, and adaptive security solutions tailored to the unique challenges of IoT ecosystems.

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol/Abbreviation	Description
PCC	PearsonCorrelationCoefficient
χ^2	Chi-squareStatistic
λ	Eigenvalue (used in PCA calculations)
σ	StandardDeviation
μ	Meanofa Dataset
Oi	ObservedFrequency(inChi- square calculations)
Ei	ExpectedFrequency(inChi- square calculations)
Xnormalized	NormalizedFeatureValue
Xmin	MinimumValueofFeature
Xmax	MaximumValueofFeature
IoT	InternetofThings
NIDS	Network Intrusion Detection System
FS	FeatureSelection
FE	FeatureExtraction
PCA	PrincipalComponentAnalysis
AE	Autoencoder
DT	DecisionTree
RF	RandomForest
kNN	k-NearestNeighbors
NB	NaiveBayes
MLP	Multi-LayerPerceptron
DDoS	DistributedDenialofService
DoS	Denialof Service
SMOTE	Synthetic Minority Over- sampling Technique
MCC	Matthews Correlation Coefficient
TON-IoT	Telemetry-Operational NetworkingIoTDataset
BoT-IoT	BotnetIoTDataset

Table :ListofSymbols

REFERENCES

[1] Overview for machine learning algorithms to enhance iot system security–https://www.nature.com/articles/s41598-024-62861-y

[2] Efficientnetworkintrusiondetection using pca-based dimensionality reduction of feature– https://ieeexplore.ieee.org/document/8909140

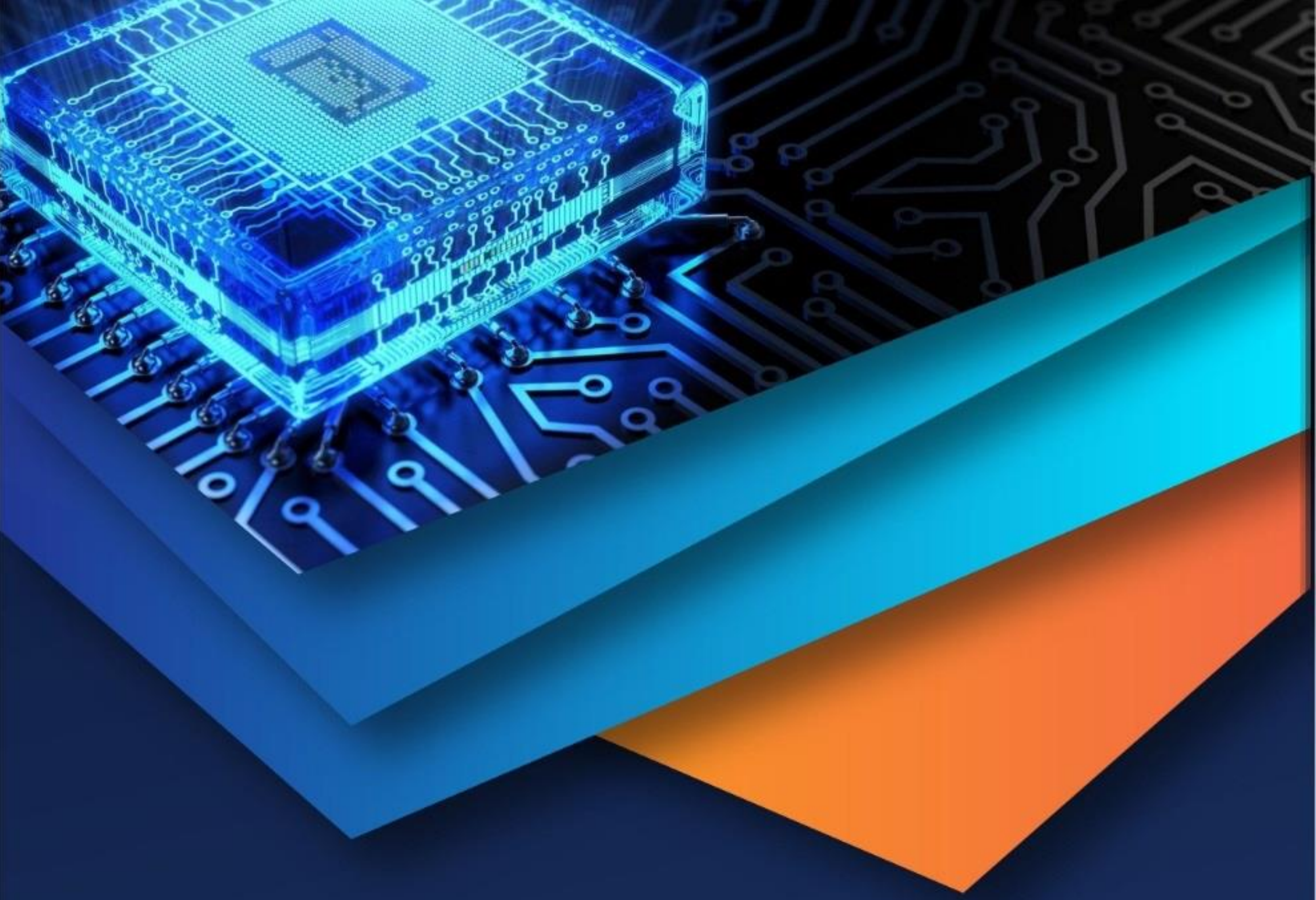
[3] Enhancing iot security through machine learning-driven anomaly detection– HTTPS://WWW.RESEARCHGATE.NET/PUBLICATION/38291 3162ENHANCINGIOTSECURITYLEVERAGIN GARTIFICIALINTELLIGENCE

[4] Anovel svm-knn-pso ensemble methodforintrusiondetectionsystem –https://linkinghub.elsevier.com/retrie ve/pii/s1568494615006328

[5] Internet of things: a survey on enabling technologies,protocols,and applications– https://ieeexplore.ieee.org/document/8080566

[6] Evaluation of machine learning algorithms for intrusion detection system–https://ieeexplore.ieee.org/document/8 080566

[7] Building an intrusion detection system using a filter-based feature selectionalgorithm– https://ieeexplore.ieee.org/document/7387736



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY