International Journal of Research in Engineering, Science and Management
Volume 7, Issue 6, June 2024
https://www.ijresm.com | ISSN (Online): 2581-5792

An Intelligent Approach to Improve Threat Detection in IoT

K. P. Shana Sherin[1*], Karibasappa Kwadiki[2], Silja Varghese[3], B. Shaji[4]
[1]PG Student, Department of Computer Science and Engineering, Nehru College of Engineering and Research Centre, Pampady, Thrissur, Kerala, India
[2]Professor, Nehru College of Engineering and Research Centre, Pampady, Thrissur, Kerala, India
[3,4]Assistant Professor, Department of Computer Science and Engineering, Nehru College of Engineering and Research Centre, Pampady, Thrissur, Kerala, India

***Abstract*: The venture expects to further develop danger discovery viability in Internet of Things (IoT) frameworks utilizing a savvy approach. IoT frameworks, which incorporate gadgets, sensors, organizations, and programming, much of the time incorporate security shortcomings that assailants could take advantage of. Utilizing ML strategies and principle component analysis (PCA), the review intends to recognize Distributed Denial of Service (DDoS) attacks, which are a typical danger to IoT gadgets. Head part investigation assists with lessening information dimensionality, smooth out datasets, and save crucial data. To actually assess model execution, assessment incorporates boundaries like as accuracy, precision, recall, and the F1-Score. The CICIDS 2017 and CSE-CIC-IDS 2018 datasets are utilized to prepare and assess the models appropriately. When contrasted with different methods, the recommended arrangement beats them and requires less preparation time, exhibiting its adequacy in further developing danger identification in IoT frameworks. We grow our exploration by utilizing ensemble approaches like Voting Classifier (RF + Adaboost) and Stacking Classifier (RF + MLP with LightGBM), bringing about a refined and exact expectation model with 100 percent accuracy. This study further develops danger recognition abilities, however it additionally shows the capability of ensemble approaches in fortifying IoT framework security.**

***Keywords*: Machine Learning, principal component analysis, Internet of Things, DDoS attack.**

1. Introduction

Industry 4.0, the Fourth Modern Unrest, changes business, assembling, and society. A combination of problematic innovations including IoT, AI, Cloud computing, and RPA supports it [1]. These center highlights have empowered unrivaled proficiency, connectedness, and computerization across businesses, introducing another time of advancement and disturbance.

IoT is a vital innovation of Industry 4.0, entering day to day existence and modern exercises [2]. Its far-reaching use empowers shrewd environmental elements, astute frameworks, and information driven direction by incorporating the physical and computerized universes. Smart homes use IoT sensors and focal regulators to robotize lighting and device the executives, working on personal satisfaction [3]. IoT can adjust medical services, horticulture, catastrophe the board, and inability support outside homes [4].

Over 13.8 billion IoT gadgets were introduced all around the world in 2021, with 30.9 billion expected by 2025 [5]. As IoT gadgets face numerous weaknesses and assaults, this quick multiplication has raised network safety concerns [6]. The ascent in IoT attacks shows danger entertainers' skill and malignance. From Q3 2019 to Q4 2020, IoT attacks expanded 3,000% and Mozi botnet commonness expanded 74% [7]. Ransomware, unlawful server access, and DDoS attacks cause most interruption [7].

A few factors make IoT frameworks powerless against cyberattacks. Security weaknesses in IoT sensor gadgets are normal in light of the fact that to unfortunate assembling [7]. Edge network parts frequently need cyberdefenses [7]. IoT networks are significant focuses for cybercriminals because of their information esteem [7].

DDoS attacks are especially harming to IoT biological systems [8]. These assaults utilize dispersed PC ability to flood target frameworks with noxious correspondences, hindering working. DDoS assaults in IoT settings have been examined and alleviated [9]. Volumetric, convention, and application-layer chases down face different insurance issues [9].

To stay away from recognition, assailants utilize cross breed assaults [9]. To distinguish and moderate DDoS attacks in IoT settings, strong cautious components and further developed abnormality identification are required [9]. Safeguarding fundamental framework and limiting interferences requires proactive observing.

IoT framework incorporation will grow all through Industry 4.0, requiring security enhancements [10]. Digital dangers should be perceived and proactive safeguards executed in light of state-of-the-art research [10]. We can keep IoT foundation changing humanity by safeguarding it.

All in all, Industry 4.0 attendants in another time of mechanical advancement and disturbance, with IoT changing society and industry. This change should be upheld by drives to get IoT conditions against developing digital dangers. We can utilize IoT while safeguarding against hazardous entertainers

*Corresponding author: shanasherinkp999@gmail.com

and getting Industry 4.0.2's future through proactive guard and partner cooperation.

2. Literature Survey

IoT has carried availability and robotization to numerous aspects of present-day life. In spite of the advantages of IoT biological systems, network protection dangers and weaknesses have expanded. Distributed Denial of Service (DDoS) attacks compromise IoT framework uprightness and working, requiring broad review to comprehend, identify, and alleviate them. This writing examination looks at current advances in IoT DDoS recognition and relief utilizing different academic papers.

Velasquez et al. [1] propose a crossover ML group for ongoing Industry 4.0 peculiarity recognition. Utilizing ML, including group draws near, the proposed framework can distinguish and moderate deviant conduct in complex modern circumstances. The innovation reinforces Industry 4.0 frameworks against cyberattacks like DDoS by joining a few information sources and utilizing progressed oddity location calculations.

Mishra and Pandya [6] investigate IoT applications, security issues, dangers, interruption location, and future yearnings. The point-by-point research shows how innovation progresses, network safety chances, and authoritative structures influence IoT security. The survey reveals insight into IoT security, especially the ascent of DDoS attacks and the requirement for proactive guards, by coordinating discoveries from an extensive variety of writing.

Da Silva Cardoso et al. [13] offer a confounded occasion handling based continuous DDoS location framework for IoT settings. Complex occasion handling's versatility and flexibility permit the framework to recognize DDoS attacks rapidly and precisely, limiting IoT interferences. The examination underlines the requirement for versatile and setting mindful location frameworks to shield IoT foundations from changing digital dangers.

Praseed and Thilagam [15] use HTTP demand design based marks to identify application-layer DDoS attacks early. The proposed arrangement rapidly recognizes and mitigates electronic DDoS attacks by breaking down HTTP demand designs. The examination accentuates the requirement for granular investigation and specific recognition strategies to forestall progressed DDoS assault vectors in IoT conditions.

You et al. [16] offer a parcel in-message-based DDoS recognition technique for SDN settings. SDN regulator parcel in messages are broke down to recognize DDoS exercises and empower speedy response and moderation. The exploration shows that network-level checking and SDN advancements further develop IoT network strength against DDoS attacks and other digital dangers.

Wehbi et al. [17] comprehensively audit ML based IoT DDoS identification strategies. The study surveys ML calculations, include choice methodologies, and recognition strategies for DDoS relief utilizing an assortment of writing. The study assists analysts and professionals with further developing IoT framework security by incorporating experimental information and logical strategies.

Maseer et al. [18] benchmark peculiarity based interruption recognition framework ML calculations utilizing CICIDS2017. The review analyzes ML models' capacity to anticipate DDoS attacks and other digital risks through thorough testing and execution assessment. The outcomes upgrade how we might interpret ML based interruption identification and IoT security.

Erhan and Anarim [20] offer a mixture DDoS discovery strategy utilizing matching pursuit. The structure recognizes IoT DDoS attacks continuously utilizing the matching pursuit calculation's registering effectiveness and flexibility. The task underlines algorithmic development and mixture identification to counter unique and various DDoS dangers in IoT organizations.

The writing study finishes up with the assortment of exploration endeavors to comprehend, distinguish, and relieve IoT DDoS attacks. From half breed ML gatherings to continuous recognition calculations for Industry 4.0 frameworks, specialists are continually growing better approaches to get IoT organizations. Researchers orchestrate exact examinations, deliberate audits, and benchmarking investigations to illuminate online protection practices and strategy systems in the period of unavoidable association and robotization.

3. Methodology

A. Proposed Work

The proposed exertion means to further develop danger identification in IoT frameworks, with a particular accentuation on recognizing and determining Distributed Denial of Service (DDoS) attacks. By consolidating ML techniques and guideline part investigation (PCA), the strategy actually prepares and predicts such attacks while improving on information through dimensionality decrease. Model execution is surveyed completely utilizing assessment measurements like as accuracy, precision, recall, and F1-Score, as well as creative estimations like Training Time. Utilizing datasets, for example, CICIDS 2017 and CSE-CIC-IDS 2018, the model's value is totally analyzed, uncovering more prominent execution and more limited preparing time than in past exploration. As an expansion, stacking (RF + MLP with LightGBM) and voting (RF + AdaBoost) classifiers are utilized to further develop danger discovery and accomplish 100 percent exactness. These group approaches further develop framework versatility by extending the discovery abilities. An easy to understand Flask interface guarantees openness, while client confirmation highlights fortify the intrusion detection system (IDS) and shield IoT conditions from unlawful access.

B. System Architecture

The task "An Intelligent Approach to Improve the Performance of Threat Detection in IoT" utilizes an orderly engineering that incorporates information handling, preparing set creation, and model structure utilizing different calculations like Random Forest [17], Decision Tree [20], Extra Tree[19], Naive Bayes, and SVM, as well as extension models such as

Voting Classifier (RF + AdaBoost) and Stacking Classifier (RF + MLP with LightGBM). The info datasets, CIC IDS 2018 and CIC IDS 2017, are preprocessed prior to being isolated into preparing and testing sets. The prepared models are then evaluated against the testing set to decide their presentation. This total framework configuration gives a full assessment and approval of the proposed keen method to further developing danger identification in IoT frameworks.

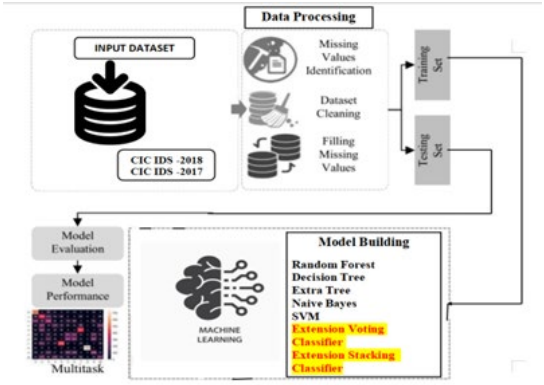

Fig. 1. Proposed architecture

C. Dataset

We utilized two datasets to assess our proposed framework: CICIDS 2017 and CSE-CIC-IDS 2018. The CICIDS 2017 dataset, delivered by the Canadian Institute for Cybersecurity (CIC), and the CSE-CIC-IDS 2018 dataset, together delivered by CSE and CIC, were picked for their appropriateness in assessing the location of Distributed Denial of Service (DDoS) assaults, which are the essential focal point of this study [1]. These datasets meet 11 Intrusion Prevention System (IPS) dataset necessities, including culmination, marked information, assault assortment, and other significant attributes [1]. Strikingly, current assault techniques were utilized in the formation of these datasets, which incorporated an assortment of organization parts like firewalls, switches, switches, and working frameworks, as well as discrete casualty and assailant zones to duplicate sensible situations.

	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	Bwd Packet Length Max	...	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min
0	640	7	4	440	358	220	0	62.857143	107.349008	179	...	0.0	0.0	0	0	0.0	0.0	0	0
1	900	9	4	600	2944	300	0	66.666667	132.287566	1472	...	0.0	0.0	0	0	0.0	0.0	0	0
2	1205	7	4	2776	2830	1388	0	396.571429	677.274651	1415	...	0.0	0.0	0	0	0.0	0.0	0	0
3	511	7	4	452	370	226	0	64.571429	110.276708	185	...	0.0	0.0	0	0	0.0	0.0	0	0
4	773	9	4	612	2944	306	0	68.000000	134.933317	1472	...	0.0	0.0	0	0	0.0	0.0	0	0

5 rows × 79 columns

Fig. 2. CIC IDS 2017 dataset

	Unnamed: 0	Dst Port	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	Fwd Pkt Len Mean	...	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle M
0	0	3389	1665875	8	7	1128	1581.0	661	0	141.00	...	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
1	1	53	67765	2	2	94	268.0	47	47	47.00	...	8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
2	2	0	213190	5	0	0	0.0	0	0	0.00	...	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
3	3	41967	86370853	2	0	0	0.0	0	0	0.00	...	20	0.0	0.0	0.0	0.0	86400000.0	0.0	86400000.0	86400000
4	4	80	5113386	4	4	97	231.0	97	0	24.25	...	20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

5 rows × 74 columns

Fig. 3. CIC IDS 2018 dataset

The proposed model was assessed utilizing specific documents from each dataset: the "Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv" record from CICIDS 2017 and the "02-21-2018.csv" record from CSE-CIC-IDS 2018, which recognized DDoS attacks [1]. The CICIDS 2017 dataset incorporates 225,745 examples and 79 elements, which were diminished to 44 highlights and 221,125 examples after information cleaning. Among them, 128,014 accounts demonstrate DDoS attacks, while 93,111 comprise harmless action. Conversely, the CSE-CIC-IDS 2018 dataset has 1,046,845 examples and 80 highlights, with 559,651 examples and 21 attributes after information cleaning, including 198,861 DDoS records and 360,790 harmless records [1]. It is essential to recollect that the element count contains names that show whether the action is pernicious or harmless.

D. Data Processing

Data processing starts with erasing copy information passages. Copies can inclination logical outcomes and cause model preparation and appraisal blunders. Copy things are taken out to keep up with dataset trustworthiness and guarantee that every information point contributes separately to examination.

Drop cleaning follows copy information evacuation. Drop cleaning eliminates pointless or unnecessary dataset qualities. Eliminate highlights that don't add to the investigation or are emphatically connected. The dataset is smoothed out to decrease dimensionality and computational intricacy while keeping up with the main credits for investigation.

Connection investigation, highlight importance positioning, and space information based determination can clear drops. Excess or superfluous qualities are taken out from the dataset by purposefully evaluating their commitment to the logical goals. This strategy keeps simply the most helpful and discriminative attributes, further developing information examination proficiency.

Data processing incorporates two essential advances: erasing copy information passages to keep up with dataset honesty and drop cleaning to eliminate immaterial qualities. The dataset should be ready for examination to guarantee right model preparation, evaluation, and understanding.

E. Visualization

Seaborn and Matplotlib help investigate and examine information by visualizing it. Dissipate plots and histograms from Seaborn uncover information dispersion and connections. utilizing more customization prospects, Matplotlib makes modern perceptions utilizing Seaborn. These instruments convey fundamental examples, exceptions, and patterns. Line charts show fleeting patterns, disperse plots variable connections. Histograms show information scattering. Seaborn's Pandas incorporation works with DataFrame representation, while Matplotlib's customisation apparatuses redo visuals. Seaborn and Matplotlib assist clients with getting information bits of knowledge for better independent direction and speculation detailing.

F. Label Encoding

Label encoding changes over clear cut factors into mathematical portrayal for ML techniques that need mathematical information sources. An all out factor's classes are given number names in this technique. This change assists

calculations with assessing absolute information. Label encoding may inadvertently give downright factors ordinality or progressive system, which could delude the calculation into giving mathematical marks inaccurate significance. Since downright factors have no characteristic request or progressive system, name encoding is typically utilized. Regardless of its straightforwardness and viability in changing clear cut information to numbers, encoded names should suitably address unmitigated factors without bias or misjudging.

G. Feature Selection

ML depends on feature selection to distinguish the most important elements for model preparation. Choosing X and y information involves distinguishing the autonomous factors (highlights) X and the reliant variable (objective) y. These factors are picked for prescient importance. Connection and shared data are ordinary element choice techniques. Straight connection looks at the connection between include matches and the objective variable. High connection values reflect critical relationships and prescient worth. Common data assesses the amount of information acquired from another variable. It catches straight and nonlinear relationships, making it more versatile in recognizing huge attributes. These strategies keep up with qualities with the most grounded prescient potential, working on model exactness and interpretability while limiting figuring intricacy.

H. Algorithms

1) Random Forest:

- With PCA: Random Forest [17] benefits from PCA in light of the fact that it lessens dimensionality, which can further develop preparing execution, decrease memory use, and breaking point overfitting.
- Without PCA: Working straightforwardly on the first element space might require extra figuring assets and time, subsequently expanding model intricacy and overfitting perils, especially with countless highlights.

Random Forest

```
from sklearn.ensemble import RandomForestClassifier

# instantiate the model
rf = RandomForestClassifier(random_state=10)

# fit the model
rf.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = rf.predict(X_test)

rf_acc = accuracy_score(y_pred, y_test)
rf_prec = precision_score(y_pred, y_test,average='weighted')
rf_rec = recall_score(y_pred, y_test,average='weighted')
rf_f1 = f1_score(y_pred, y_test,average='weighted')
```

```
storeResults('Random Forest',rf_acc,rf_prec,rf_rec,rf_f1)
```

Fig. 4. Random Forest

2) Decision Tree

- With PCA: Decision Trees [20] work on a more modest element space, potentially improving on the tree structure and upgrading interpretability.
- Without PCA: Decision Trees can create convoluted trees with various attributes, raising the risk of overfitting by gathering clamor in the information.

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier

# instantiate the model
tree = DecisionTreeClassifier(max_depth=30)

# fit the model
tree.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = tree.predict(X_test)


dt_acc = accuracy_score(y_pred, y_test)
dt_prec = precision_score(y_pred, y_test,average='weighted')
dt_rec = recall_score(y_pred, y_test,average='weighted')
dt_f1 = f1_score(y_pred, y_test,average='weighted')
```

```
storeResults('Decision Tree',dt_acc,dt_prec,dt_rec,dt_f1)
```

Fig. 5. Decision Tree

3) Extra Tree (Extremely Randomized Trees)

- With PCA: Extra Trees [19], similar to Random Forest, benefit from PCA by working on a more modest element space, perhaps expanding productivity and bringing down overfitting concerns.
- Without PCA: Working straightforwardly on the first elements might bring about lengthier preparation periods and a bigger gamble of overfitting, especially for high-layered information.

ExtraTree

```
from sklearn.ensemble import ExtraTreesClassifier

# instantiate the model
et = ExtraTreesClassifier(random_state=10)

# fit the model
et.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = et.predict(X_test)

et_acc = accuracy_score(y_pred, y_test)
et_prec = precision_score(y_pred, y_test,average='weighted')
et_rec = recall_score(y_pred, y_test,average='weighted')
et_f1 = f1_score(y_pred, y_test,average='weighted')
```

```
storeResults('ExtraTree',et_acc,et_prec,et_rec,et_f1)
```

Fig. 6. Extra Tree

4) Naive Bayes

Naive Bayes

```
from sklearn.naive_bayes import GaussianNB

# instantiate the model
nb = GaussianNB()

# fit the model
nb.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = nb.predict(X_test)

nb_acc = accuracy_score(y_pred, y_test)
nb_prec = precision_score(y_pred, y_test,average='weighted')
nb_rec = recall_score(y_pred, y_test,average='weighted')
nb_f1 = f1_score(y_pred, y_test,average='weighted')
```

```
storeResults('Naive Bayes',nb_acc,nb_prec,nb_rec,nb_f1)
```

Fig. 7. Naive Bayes

- With PCA: PCA can decrease commotion and computational expense in high-layered datasets, but since Naive Bayes expects highlight freedom, it might not considerably affect its exhibition.
- Without PCA: Naive Bayes works straightforwardly on the first information and may battle with high-layered datasets and computational intricacy.

5) Support Vector Machine (SVM)

- With PCA: PCA helps SVM by bringing down how much elements, which might further develop speculation and registering effectiveness by working in a more modest component space.
- Without PCA: Working on the first element space without PCA might bring about computational requests and overfitting dangers, particularly for huge quantities of highlights.

SVM

```
from sklearn.svm import SVC

# instantiate the model
svm = SVC(probability=True)

# fit the model
svm.fit(X_train, y_train)

#predicting the target value from the model for the samples

y_pred = svm.predict(X_test)

svc_acc = accuracy_score(y_pred, y_test)
svc_prec = precision_score(y_pred, y_test,average='weighted')
svc_rec = recall_score(y_pred, y_test,average='weighted')
svc_f1 = f1_score(y_pred, y_test,average='weighted')
```

```
storeResults('SVC',svc_acc,svc_prec,svc_rec,svc_f1)
```

Fig. 8. Support Vector Machine (SVM)

6) Voting Classifier

The Voting Classifier joins a few base assessors (Random Forest and AdaBoost in this model) and midpoints their forecasts by means of voting.

Voting Classifier might utilize the capacities of Random Forest and AdaBoost to increment danger discovery execution in IoT settings. It can gather a few components of information and work on the general versatility of the recognizing framework.

Voting Classifier

```
from sklearn.ensemble import RandomForestClassifier, VotingClassifier, AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier

rfc = RandomForestClassifier()
parameters = {
    "n_estimators":[250],
    "max_depth":[200]

}

from sklearn.model_selection import GridSearchCV
forest = GridSearchCV(rfc,parameters,cv=10)

clf2 = DecisionTreeClassifier(random_state=1000)

eclf1 = VotingClassifier(estimators=[('rf-parameter', forest), ('dt', clf2)], voting='soft')
eclf1.fit(X_train, y_train)
y_pred = eclf1.predict(X_test)

vot_acc = accuracy_score(y_pred, y_test)
vot_prec = precision_score(y_pred, y_test,average='weighted')
```

Fig. 9. Voting classifier

7) Stacking Classifier

The Stacking Classifier prepares a few base assessors (Random Forest and Multi-Layer Perceptron with LightGBM) and afterward utilizes a meta-student to join their expectations.

The Stacking Classifier might consolidate the prescient capacity of Random Forest, a neural network (MLP), and LightGBM to build a further developed model for threat identification. By coordinating the qualities of numerous calculations, it can deal with confounded examples and increment danger location exactness in IoT settings.

Stacking Classifier

```
from sklearn.neural_network import MLPClassifier
from lightgbm import LGBMClassifier
from sklearn.ensemble import StackingClassifier

estimators = [('rf', forest),('mlp', MLPClassifier(random_state=1, max_iter=3000))]

clf = StackingClassifier(estimators=estimators, final_estimator=LGBMClassifier(n_estimators=1000))


clf.fit(X_train,y_train)

y_pred = clf.predict(X_test)

stac_acc = accuracy_score(y_pred, y_test)
stac_prec = precision_score(y_pred, y_test,average='weighted')
stac_rec = recall_score(y_pred, y_test,average='weighted')
stac_f1 = f1_score(y_pred, y_test,average='weighted')
```

```
storeResults('Stacking Classifier',stac_acc,stac_prec,stac_rec,stac_f1)
```

Fig. 10. Stacking classifier

4. Experimental Results

1) Accuracy: Accuracy is defined as the proportion of correct forecasts in a grouping location, which represents a model's overall accuracy.

Accuracy = (True Positive + True Negative)/(True Positive + True Negative + False Positive + False Negative)

2) F1-Score: The F1 score is a symphonic mechanism of precision and validation that is proper for imbalanced datasets since it catches both false positives and false negatives.

F1-Score = (2 * Precision * Recall)/ (Precision + Recall)

3) Precision: Precision estimates the extent of plainly characterized cases or tests among those classed as certain. Consequently, the precision might be registered with the accompanying equation:

Precision = True Positive / (True Positive + False Positive)

4) Recall: Recall is a ML measurement that surveys a model's capacity to perceive all occasions of a particular class. It is the negligible part of accurately anticipated positive perspectives that amount to actual advantages, which gives data on a model's outcome in gathering instances of a particular class.

Recall = True Positive + False Negative

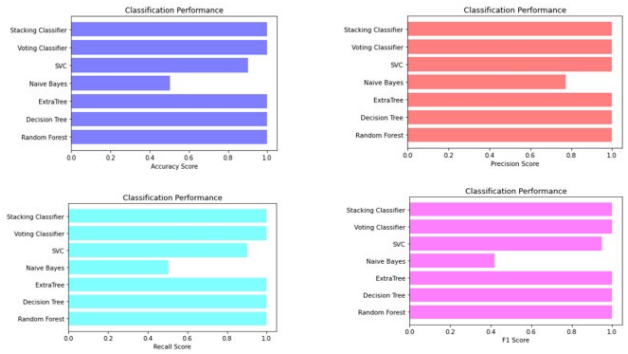

Fig. 11. Comparison Graphs of CIC – IDS – 2017 Dataset (Without PCA)

Table 1
Comparison Graphs of CIC – IDS – 2018 Dataset (With PCA)

ML Model	Accuracy	Precision	Recall	F1-Score
Random Forest	1.000	1.000	1.000	1.000
Decision Tree	1.000	1.000	1.000	1.000
Extra Tree	1.000	1.000	1.000	1.000
Naïve Bayes	0.503	0.772	0.503	0.419
SVC	0.901	1.000	0.901	0.948
Extension Voting Classifier	1.000	1.000	1.000	1.000
Extension Stacking Classifier	1.000	1.000	1.000	1.000

Table 2
Performance Evaluation Table CID – IDS 2017 Dataset

PCA ML Model	Accuracy	Precision	Recall	F1-Score
PCA Random Forest	1.000	1.000	1.000	1.000
PCA Decision Tree	0.998	0.998	0.998	0.998
PCA Extra Tree	1.000	1.000	1.000	1.000
PCA Naïve Bayes	0.494	0.773	0.494	0.410
PCA SVM	0.901	1.000	0.901	0.948
Extension PCA Voting Classifier	0.999	0.999	0.999	0.999
Extension PCA Stacking Classifier	1.000	1.000	1.000	1.000

Table 3
Performance Evaluation Table CID – IDS 2017 Dataset (PCA)

ML Model	Accuracy	Precision	Recall	F1-Score
Random Forest	1.000	1.000	1.000	1.000
Decision Tree	1.000	1.000	1.000	1.000
Extra Tree	1.000	1.000	1.000	1.000
Naive Bayes	0.594	0.924	0.594	0.676
SVC	0.575	0.937	0.575	0.670
Extension Voting Classifier	1.000	1.000	1.000	1.000
Extension Stacking Classifier	1.000	1.000	1.000	1.000

Table 4
Performance Evaluation Table CID – IDS 2018 Dataset

PCA ML Model	Accuracy	Precision	Recall	F1-Score
PCA Random Forest	0.999	0.999	0.999	0.999
PCA Decision Tree	0.999	0.999	0.999	0.999
PCA Extra Tree	0.999	0.999	0.999	0.999
PCA Naïve Bayes	0.603	0.918	0.603	0.619
PCA SVM	0.575	0.937	0.575	0.670
Extension PCA Voting Classifier	1.000	1.000	1.000	1.000
Extension PCA Stacking Classifier	0.999	0.999	0.999	0.999

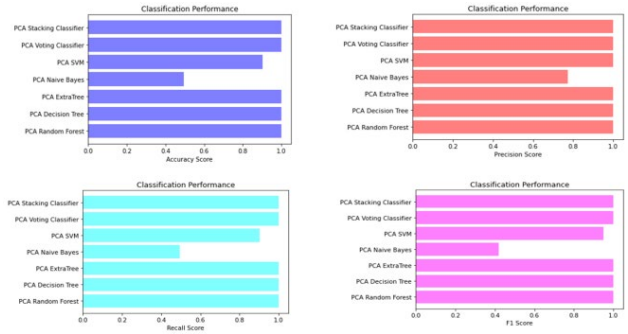

Fig. 12. Comparison Graphs of CIC – IDS – 2017 Dataset (With PCA)

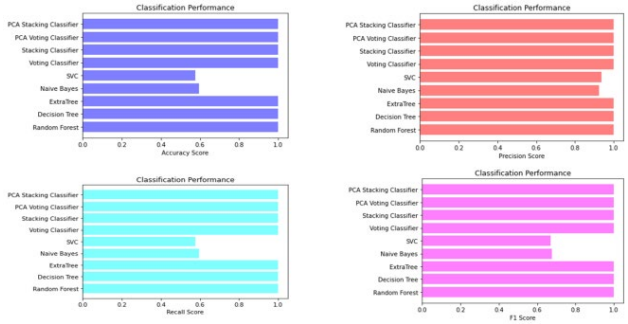

Fig. 13. Comparison Graphs of CIC – IDS – 2018 Dataset (Without PCA)

5. Conclusion

This paper presented an intelligent approach to improve threat detection in IoT.

References

[1] D. Velasquez, E. Perez, X. Oregui, A. Artetxe, J. Manteca, J. E. Mansilla, M. Toro, M. Maiza, and B. Sierra, "A hybrid machine-learning ensemble for anomaly detection in real-time industry 4.0 systems," IEEE Access, vol. 10, pp. 72024–72036, 2022.

[2] S. U. Rehman and V. Gruhn, "An approach to secure smart homes in cyber-physical systems/Internet-of-Things," in Proc. 5th Int. Conf. Softw. Defined Syst. (SDS), Barcelona, Spain, Apr. 2018, pp. 126–129.

[3] S. K. Vishwakarma, P. Upadhyaya, B. Kumari, and A. K. Mishra, "Smart energy efficient home automation system using IoT," in Proc. 4th Int. Conf. Internet Things, Smart Innov. Usages (IoT-SIU), Ghaziabad, India, Apr. 2019, pp. 417–420.

[4] S. Chaudhary, R. Johari, R. Bhatia, K. Gupta, and A. Bhatnagar, "CRAIoT: Concept, review and application(s) of IoT," in Proc. 4th Int. Conf. Internet Things, Smart Innov. Usages (IoT-SIU), Ghaziabad, India, Apr. 2019, pp. 402–405.

[5] Lionel Sujay Vailshery. 2022, [Online]. Available: https://www.statista.com

[6] N. Mishra and S. Pandya, "Internet of Things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," IEEE Access, vol. 9, pp. 59353–59377, 2021.

[7] X-Force Threat Intelligence Index 2022, IBM Security, Atlanta, GA, USA, 2022.
[8] D. Patel, "A study on DDOS attacks, danger and its prevention," Int. J. Res. Appl. Sci. Eng. Technol., vol. 10, no. 12, pp. 1962–1967, Dec. 2022.
[9] N. Vlajic and D. Zhou, ''IoT as a land of opportunity for DDoS hackers,'' Computer, vol. 51, no. 7, pp. 26–34, Jul. 2018.
[10] T. U. Sheikh, H. Rahman, H. S. Al-Qahtani, T. K. Hazra, and N. U. Sheikh, "Countermeasure of attack vectors using signature-based IDS in IoT environments," in Proc. IEEE 10th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON), Vancouver, BC, Canada, Oct. 2019, pp. 1130–1136.
[11] R. Zhang, J.-P. Condomines, N. Larrieu, and R. Chemali, "Design of a novel network intrusion detection system for drone communications," in Proc. IEEE/AIAA 37th Digit. Avionics Syst. Conf. (DASC), London, U.K., Sep. 2018, pp. 241–250.
[12] F. Suthar, N. Patel, and S. V. O. Khanna, "A signature-based botnet (Emotet) detection mechanism," Int. J. Eng. Trends Technol., vol. 70, no. 5, pp. 185–193, May 2022.
[13] A. M. da Silva Cardoso, R. F. Lopes, A. S. Teles, and F. B. V. Magalhaes, "Poster abstract: Real-time DDoS detection based on complex event processing for IoT," in Proc. IEEE/ACM 3rd Int. Conf. Internet-Things Design Implement. (IoTDI), Orlando, FL, USA, Apr. 2018, pp. 273–274.
[14] M. Dimolianis, A. Pavlidis, and V. Maglaris, "Signature-based traffic classification and mitigation for DDoS attacks using programmable network data planes," IEEE Access, vol. 9, pp. 113061–113076, 2021.
[15] A. Praseed and P. S. Thilagam, "HTTP request pattern based signatures for early application layer DDoS detection: A firewall agnostic approach," J. Inf. Secur. Appl., vol. 65, Mar. 2022, Art. no. 103090.
[16] X. You, Y. Feng, and K. Sakurai, "Packet in message based DDoS attack detection in SDN network using OpenFlow," in Proc. 5th Int. Symp. Comput. Netw. (CANDAR), Nov. 2017, pp. 522–528.
[17] K. Wehbi, L. Hong, T. Al-salah, and A. A. Bhutta, "A survey on machine learning based detection on DDoS attacks for IoT systems," in Proc. SoutheastCon, Huntsville, AL, USA, Apr. 2019, pp. 1–6.
[18] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa and C. F. M. Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," in IEEE Access, vol. 9, pp. 22351-22370, 2021.
[19] N. T. Cam and N. G. Trung, "An Intelligent Approach to Improving the Performance of Threat Detection in IoT," in IEEE Access, vol. 11, pp. 44319-44334, 2023.
[20] M. Najafimehr, S. Zarifzadeh, and S. Mostafavi, "A hybrid machine learning approach for detecting unprecedented DDoS attacks," J. Supercomput., vol. 78, no. 6, pp. 8106–8136, Apr. 2022.
[21] D. Erhan and E. Anarim, "Hybrid DDoS detection framework using matching pursuit algorithm," IEEE Access, vol. 8, pp. 118912–118923, 2020.