International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

# Efficient Trust Inference Model for Pervasive Computing Based on Hybrid Deep Learning

Mrs. Geetha Pawar[1*], Dr. Jayashree Agarkhed[2]

**Submitted:** 11/11/2022 **Accepted:** 14/02/2023

**Abstract*:*** With the rise of mobile technology, pervasive computing has become an indispensable tool for processing and exchanging data in today's connected society. For distributed computing services to be used in the same places where people live and socialise, pervasive computing must be present. Recent developments in pervasive computing have shifted the focus from stationary computers to mobile ones, such as laptops, notepads, cell phones, and PDAs. Devices in the pervasive environment are available on a global scale and are capable of receiving a wide range of audio-visual as well as other telecommunications services. Problems of user trust, data protection, and client and device node identification may arise for the system and users in this extensive setting. In this study, we proposed an efficient trust inference model for ubiquitous computing associated fine-tuned ANN and ML IoT attack predictor which reached an accuracy of 90.43%.

***Keywords:*** *Anomaly Detection, Machine Learning, Security, Pervasive Computing, Enhanced Trust model.*

## 1. Introduction

The Internet of Things (IoT) has recently received significant interest in the IT sector, and for a good reason. As computing power and data storage needs grow, people are increasingly turning away from bulky desktop PCs in favour of more portable devices that can handle complex tasks [1]. As the newest concept in computer science, ubiquitous computing is revolutionising the industry.

Pervasive technology encourages the incorporation of intelligent gadgets into our living areas so that we can design novel solutions using such gadgets [2][3]. These gadgets, which may include sensing devices, actuators, and computer capacities, make it possible to construct a plethora of facilities that help people, automate the maintenance of infrastructures, or drive manufacturing operations via data analytics, among other things. There are currently a great number of these types of services available, and they are of noteworthy significance in a wide variety of industries and organisations. When it comes to computer security, the tried-and-true methods of access control and verification are what have always been relied upon to grant privileges to a few authorised users [4]. Solutions like these are not a good fit for the highly versatile and scalable computing platforms used in ubiquitous and pervasive computing [5][6]. An essential feature of ubiquitous computing is the creation and architecture of applications optimised for the client who makes the service call and the environment within which the request is made.

The application of ML strategies has recently given rise to the development of more intelligent service models [7]. The purpose of ML and deep learning is to teach a computer to function without human intervention by discovering latent trends in the information that cannot be predicted or modelled by a system of regulations. Artificial intelligence technologies have proved very effective in areas such as computer vision, natural language processing, and decision-making. As a result, it should come as no surprise that there is a growing need for using ML techniques in ubiquitous fields where conventional approaches cannot be applied due to the absence of modelling software or prohibitive computational intricacy [8].

The suggested study's objective is the creation of an effective deep learning system to guarantee fundamental safety concerns like trustworthiness, confidentiality, and integrity in the online world. The dataset was utilised in this study because of its highly rated features. Different standard ML approaches were employed before our proposed technique for this research. In this study, we described all those techniques, and the results show that our suggested approach surpasses the others with an efficiency as high as 90.43%.

[1*] *Assistant Professor, Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bengaluru, Karnataka, India.*

[2] *Professor, Department of Computer Science and Engineering, Poojya Doddappa Appa College of Engineering, Gulbarga, Karnataka, India.*

.
* *Corresponding Author Email: profgeethapawar0101@gmail.com*

## 2. Key Issues

The biggest problem with ubiquitous computing is cyber security because no one will trust their sensitive, confidential, and mission-critical data to an architecture they don't know or have reason to believe is trustworthy. The employment of many 'conflicting' encryption techniques and the existence of inherent flaws in particular wireless security mechanisms, such as those used in wireless LANs, contribute to the safety problems plaguing wireless and mobile systems.

Numerous factors combine to explain why widespread usage of robust security measures is lacking in the wireless infrastructure we rely on today. Secrecy, authenticity, authorisation, licensing, availability, and non-repudiation are only a few of several security concerns in the pervasive environment.

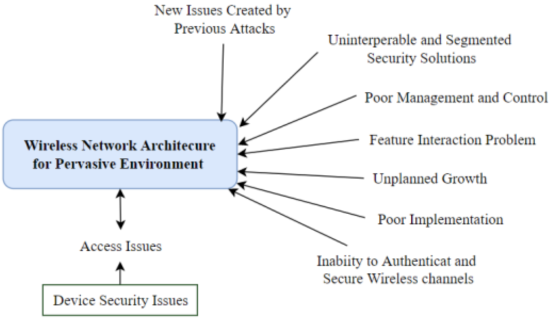

New Issues Created by Previous Attacks
Uninterperable and Segmented Security Solutions
Poor Management and Control
Feature Interaction Problem
Wireless Network Architecure for Pervasive Environment
Unplanned Growth
Poor Implementation
Access Issues
Inabiity to Authenticat and Secure Wireless channels
Device Security Issues

**Fig. 1:** Security Issues in Ubiquitous Computing

The potential cost of data destruction, alteration, or theft should inform the development and implementation of a security policy. Wireless devices expose users to additional dangers in adding to the pre-existing privacy and security threats. Multiple wireless networks, each with multiple possible degrees of security, may be part of the wireless architecture. A disruption in any of these areas could result in a loss of data or the inability to access the system altogether. Figure 1 illustrates various security issues appearing in a pervasive computing environment. The following are examples of the various attacks that can be launched against ubiquitous computing systems.

*A. Man-in-the-middle Attack*

It is crucial to authenticate devices in order to provide services in a Pervasive Computing setting. An item's authenticity is verified only after the user enters a passcode, such as a password or PIN, into a verification system. Whenever objects or people pose as third parties and relay requests and responses, they are engaging in a Man-in-the-middle assault [9].

*B. Denial Of Service (DOS)*

The goal of this kind of assault is to prevent legitimate users from accessing the targeted services and materials [10]. For instance, this may be done by preventing communication altogether, reducing services, using up all of the device's resources, or giving erroneous confidence evaluations to the various communicating nodes. As a result, it is crucial to plan and take the necessary measures to identify these circumstances.

*C. Eavesdropping Attack*

This form of attack refers to a category of assaults where a malicious attacker either observes dialogue in order to gain information regarding private information or interacts with the communications platform to alter texts [11]. In both cases, the goal is to gain access to private data. Shoulder surfing is one kind of assault. In this approach, the opponent sees the information displayed on a smart phone, surreptitiously peering over the user's shoulder as they use the gadget. Another illustration of this would be spying, in which an enemy observes in covert fashion what another individual is writing on his or her computer [12].

*D. Cryptanalytic Attack*

Passcode-breaking attacks, side-channel attacks like acoustic cryptographic algorithmic keyword search and electromagnetic breaches, and other cryptographical attacks like preimage, cipher text, birth date, and key generation intrusions are all common forms of attack [13].

*E. Access Network Attack*

The access network is what links the residential access points to the networks of the external service providers [14]. If an adversary can assemble confidential information such as financial data, user ID, and other details from a network packet when connected to a user's internal network, then it is apparent that the confidential material could be revealed.

*F. Illegal Connection Attack*

Many home appliances can be linked to several networks with a single home portal. Normally, they are managed through web-based admin, however the difficulty emerges when the adversary accesses this administrative credential. If an adversary receives this data, he can quickly launch a network-wide assault [15]. The assailants also pose as trusted employees in order to take command of domestic appliances. A disclosure of sensitive information could result in inappropriate use.

*G. Capturing Sensitive Data*

Due to their limited processing capabilities, digital sensors are widely deployed in the pervasive control process; yet, any adversary who gets a transmitter proximate enough to a sensor can steal its confidential material [16]. Rather than focusing on cryptography matters, these detectors often perform sensing function.

*H. Stealing Intermediary Device*

The sensor information is often gathered by a third-party gadget. As soon as the device falls into the wrong hands, it becomes useless because it is now part of the compromised system and can be exploited in attacks [17]. The equipment may be compromised due to the presence of a service port.

*I. Data Manipulation*

Since sensors have limited processing power, they rely on a middle device to keep track of the data they collect as they travel across a network [18]. Data verification by encrypting and decrypting still leaves room for data manipulation. With this tactic, the assailant poses as an insider and switches out real gadgets with fraudulent ones. In order for a malicious actor to steal confidential information from the system by posing as a legitimate user [19].

## 3. Literature Review

Scholars come up with a variety of ideas for methodologies and methods that can be used for ubiquitous computing. When it comes to backing up context-aware apps, Chen et al. developed a comprehensive architecture. The context-aware information sources are handled as streaming providers by the envisaged gateway [20]. The system is reliable enough to back the information-based services of a self-organizing mentoring over lays. A method for emulating ubiquitous systems based on approximate understanding of the circumstances and entities involved was developed by Katsiri and Mycroft [21]. As a result of this study's efforts, AESL now presents a greater predicate operator, which can be used to indicate an assess of one's understanding on the likelihood of a forecast occurrence having a value of True at Padovitz et al. introduced the ECORA architecture for scalable, user-friendly, communicative, and heterogeneous computation with the goal of addressing the challenges of processing in unpredictable contexts [22]. The model considers an advisor-oriented hybrid strategy and integrating provision of centralised argumentation with situationally-aware, rationale mobile software entities. The S-MARKS architecture and solution were presented by Ahamed et al., and they cover topics including information retrieval, gadget authentication, resource exploration, and modular confidentiality [23].

A secure approach for maintaining trust was developed by Boukerche and Ren [24], which includes creating a security framework, giving out credentials to endpoints, changing the secret key, keeping track of the trustworthiness, and making the appropriate call when it comes to who has exposure to what. This study proved that security flaws may be successfully removed from ubiquitous computing ecosystems. In order to develop a contrast and categorization system for the aspects of geographical, chronological, item, as well as other issues in deployment, Yu et al. combed the literature [25]. Based on many literatures review employing approaches and basic conceptualization stages for the maintenance of confidence, Usman and Gutierrez expanded the fundamental notion of ubiquitous and portable technology [26]. The research examined numerous confidence protocol methodologies, strategies, frameworks, and implementations. Carullo et al. introduced a novel method of establishing trust that makes advantage of clients' information [27]. The method proposed by Denko et al. is able to assess the reliability and behaviour of an interactive gadget with minimal user training [28]. As a means of helping academic communities construct an effective teaching platform, Ivanova reviews the present study in the area of gesture detection and face expression from the standpoint of smart mentoring [29].

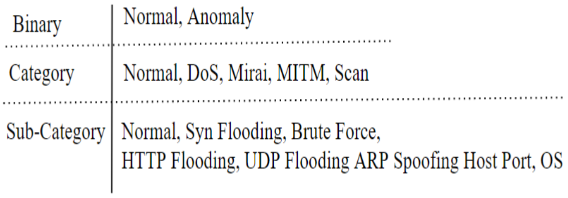
According to the research by Chang et al. quantitative aspects of data transmission that are reliant on the host system can be used to forecast when a network is breached [30]. An innovative ML strategy based on random forest and SVM was created to provide predictions about network infiltration. According to Tariq et al., a situationally-aware co-design centred intellectual safety approach in integrated IIoT contexts pays off in the form of new insights into strategy implementation, which in turn motivates further research. Their suggested system's innovation lies in its capacity to function despite erratic system connectivity and node constraints such as sparse computing capacity and a small buffering [31].

## 4. Dataset, Data Analysis and Data Pre-Processing

A high-quality dataset that can help categorise entities as accurately as probable is needed to evaluate the efficacy of contemporary intrusion detection models and techniques. Due to this reason, we utilised the IoTID20 dataset in our experimental study.

*A. Dataset*

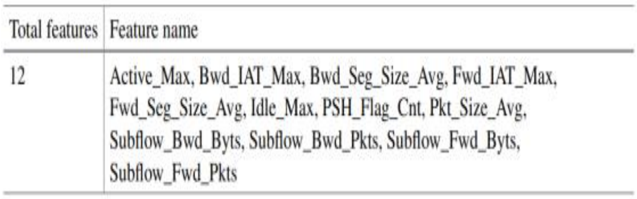
The dataset in use here is the IoTID20 dataset which was created by Ullah and Mahmoud (2020) [32]. This dataset was obtained with the help of the SKT NGU intelligent home gadget and the EZVIZ Wi-Fi camera from standard components of a smart home ecosystem. There are a total of 80 network attributes and 3 labelled attributes in the dataset. The dataset consists of three label features: the binary, the main category, and a further sub-category, presented in detail in figure 2.



| | |
|---|---|
| Binary | Normal, Anomaly |
| Category | Normal, DoS, Mirai, MITM, Scan |
| Sub-Category | Normal, Syn Flooding, Brute Force,<br>HTTP Flooding, UDP Flooding ARP Spoofing Host Port, OS |

**Fig. 2:** Categorisation of IoTID20 Dataset

A ranking of the characteristics in the dataset was performed utilising the Shapiro-Wilk technique. This technique evaluates the uniformity of the incidence dispersion regarding the attribute. Greater than 70% of features scored higher than 0.50 on the ranking scale. Incorporating these highly rated characteristics into recognition systems and algorithms can boost their capacity for categorisation. These elevated variables cannot only enhance the recognition capabilities of the ML algorithms but also facilitate the feature selection procedures, which helps reduce the amount of time required for model training. The total amount of instances in this dataset is 625783.



| Total features | Feature name |
|---|---|
| 12 | Active_Max, Bwd_IAT_Max, Bwd_Seg_Size_Avg, Fwd_IAT_Max, Fwd_Seg_Size_Avg, Idle_Max, PSH_Flag_Cnt, Pkt_Size_Avg, Subflow_Bwd_Byts, Subflow_Bwd_Pkts, Subflow_Fwd_Byts, Subflow_Fwd_Pkts |

**Fig. 3:** Correlated Features

The correlated features of this dataset are presented in figure 3.

*B. Exploratory Data Analysis*

We conducted exploratory data analysis to find out more details related to the data. The binary labels consist of two categories: Anomaly and Normal. We can see from the label distribution in figure 4 that only 6.40% of the whole data belongs to the 'Normal' category; the rest falls into the 'Anomaly' category. This indicates that 585710 packets of data are of the 'Anomaly' category.

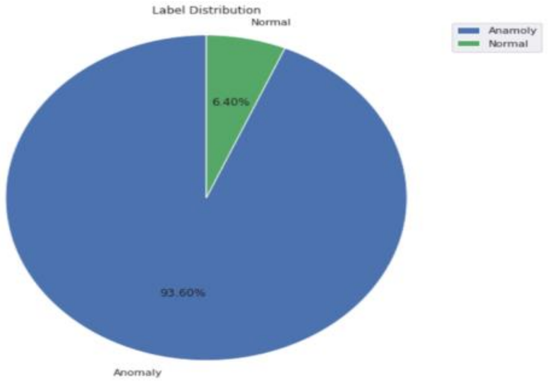


**Fig. 4:** Frequency of Binary Label Features

The IoTID20 Dataset is distributed into five main categories: 'Mirai', 'Scan', 'DoS', 'Normal', and 'MITM ARP Spoofing'. The main category distribution graph illustrated in figure 5 shows that most 'Anomaly' cases belong to the 'Mirai' class with a number of 415677 data. The 'MITM ARP Spoofing' class holds the least amount of data.

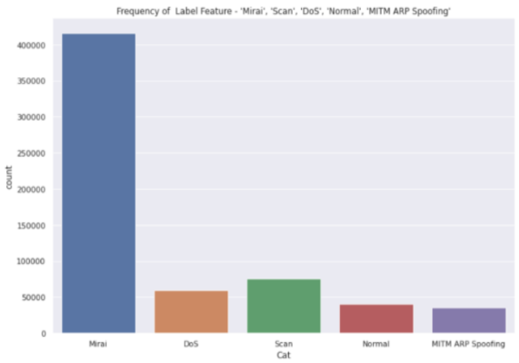


**Fig. 5:** The Distribution of Five Main Categories of the IoTID20 Dataset

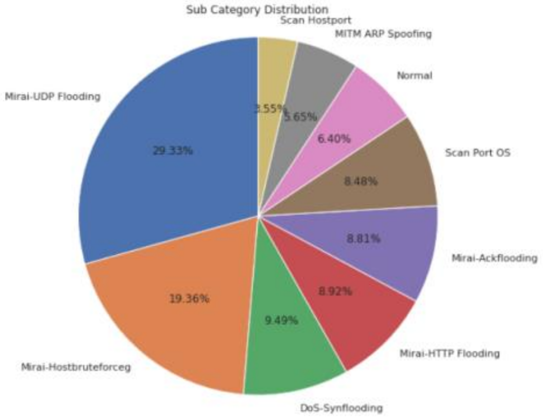


**Fig. 6:** The Sub-category Distribution

Figure 6 depicts the distribution of 8 sub-categories of the 'Anomaly' class along with the 'Normal' class in the IoTID20 Dataset. These sub-categories are described below.

1) DoS-Synflooding
2) MiraiAck Flooding
3) Mirai Brute force
4) Mirai HTTP Flooding
5) Mirai UDP Flooding
6) MITM
7) Scan Host Port
8) Scan Port OS

According to the pie-chart distribution presented in figure 6, 'Mirai UDP Flooding' class is the largest shareholder in 'Anomaly' class. Overall, four types of 'Mirai' anomaly as well as two types of ‘Scan’ anomaly can be found in the dataset. ‘Scan Host Port’ consists of least amount of data.

From further analysing the dataset we found out that, most of the information were accumulated from 3 IP sources.

### C. Data Pre-processing

The dataset needs to be pre-processed since some characteristics do not conform to standard data types and formats for usage with ML and deep learning techniques. There were no missing data in this database. So, spontaneously, we went for feature creation which can help in efficient model prediction. The created features are all time-based attributes. Later, the redundant feature columns are removed from the dataset. Some sub-categories had to be removed for the same reason.

These sub-categories are:

‘Src_IP’, ‘Src_Port’,

‘Dst_IP’, ‘Dst_Port’,

‘Protocol’, ‘Timestamp’.

Categorical data columns are converted to numerical ones for easy handling of the data.

After this comes normalisation; through normalisation, data is transformed and resampled so that they fall neatly within the interval from 0 to 1. “Min-Max scaling” is also commonly used to describe this technique. Equation 1 represents the mathematical formula of Min-Max scaling for data normalisation.

$$X' = \frac{X - X_{\text{minimum}}}{X_{\text{maximum}} - X_{\text{minimum}}} \quad \textbf{(1)}$$

After normalisation is done, the processed dataset is merged with the numeric dataset. The sub-categories are coded numerically so that it is easier for classification. The coding and related sub-categories are presented in the table 1.

**Table 1:** Coding of Sub-categories

| Sub-category | Code |
|---|---|
| 'Mirai-UDP Flooding' | 1 |
| 'Mirai-Host brute force' | 2 |
| 'DoS-Synflooding' | 3 |
| 'Mirai-HTTP Flooding' | 4 |
| 'Mirai-Ackflooding' | 5 |
| 'Scan Port OS' | 6 |
| 'MITM ARP Spoofing'<br>‘Scan Hosrport’ | 7 |
| 'Normal' | else |

### D. Data Segmentation

After the pre-processing step, data segmentation of the whole dataset is done. The while dataset is split into training and testing sets. The training dataset is assigned 80% of the data and the testing set got 20% of the data. The train_test_split () function is applied to achieve data segmentation. By using the ‘stratify’ function, we can guarantee that the same number of samples from each class appear in both the training and testing sets.

## 5. Ml and Deep Learning Models for Pervasive Computing

In this research, various ML classifiers were used on a cleaned and segmented dataset to calculate the precision and recall of each individual instance. Scikit-Learn is used for model designing here.

### A. Random Forest Classifier

During training, the random forest method creates a forest of ‘basic’ decision trees and uses the ones with the highest average accuracy to make its classifications. Through the use of a voting mechanism, the problem of decision trees overfitting training data is solved. During the training phase, random forests discard random trees. Bagging selects and fits trees using a randomly sampled, randomised, and resampled training set. Table 2 shows the parameters and corresponding values for random forest classifier here.

**Table 2:** Parameter and Values set for Random Forest Classifier

| Parameter | Value |
|---|---|
| random_state | 101 |
| n_estimators | 6 |
| n_jobs | -1 |
| max_depth | 5 |

### B. XG Boost Classifier

Boosting is an ensemble modelling technique that takes multiple weak classifiers and uses them to create a single, highly accurate one. To increase a signal, many people turn to Gradient Boosting. Each successive XGBoost predictor fixes the shortcomings of the one before it. This type of boosting method uses gradient-boosted decision trees. This algorithm constructs decision trees in stages, assigning weights based on their relative importance. The process is finished after weights are applied to all independent variables. When the outcomes of a factor were incorrectly anticipated in a prior decision tree, that factor is provided towards the next decision tree alongside instructions to improve its outcomes. Classification methods and predictors are combined to provide a more precise model. Although random fitting has two major flaws (high bias and low variance), XGBoost fixes both by giving you a wide range of tuning parameters to fine-tune so that you can build a machine learning model that will last. Table 3 shows the parameters and corresponding values for XGBoost classifier here.

**Table 3**: Parameter and Values set for XGBoost Classifier

| Parameter | Value |
|---|---|
| random_seed | 43 |
| n_estimators | 20 |
| n_jobs | -1 |
| max_depth | 10 |
| scoring | ‘f1’ |

### C. *SVM Kernel Classifier*

The objective of the SVM system is to obtain the finest border or verdict borderline which can split an n-dimensional region into categories. This can consent users to easily reside any fresh data points in the relevant group hereafter. This optimal borderline is called a hyperplane. A group of mathematical operations known as the kernel is used by SVM algorithms. The job of the kernel is to accept whatever data is fed into it and reformat it in some suitable way. The kernel functions used by each SVM method are distinct. Many kinds of capabilities can serve these purposes. Table 4 shows the parameters and corresponding values for SVM kernel classifier here.

**Table 4**: Parameter and Values set for SVM Kernel Classifier

| Parameter | Value |
|---|---|
| kernel | ‘rdf’ |

### D. *Associated Fine turned Artificial Neural Network & Machine Learning IoT Attack Predictor*

We designed a novel model architecture for identifying 9 different classifications of IoT attacks. A neural network is composed of several layers, each of which is responsible for a particular task. The number of layers in a neural network grows in proportion to the intricacy of the underlying system. This type of model takes the inputted data and then pushes those data into a group of layers. A loss function ought to be utilised for the purpose of assessing the throughput of the system. The loss function provides the network with a notion of the route it requires to pursue in order to become knowledgeable about the topic at hand [33]. The network requires the assistance of an optimiser in order to boost its level of understanding.

The model architecture is a customised ANN framework which has two stages. TensorflowKeras is used here in our study for IoT Attack Predictor model building. 50024 data packets were used in this case for model training. 12506 data packets were assigned for model testing.

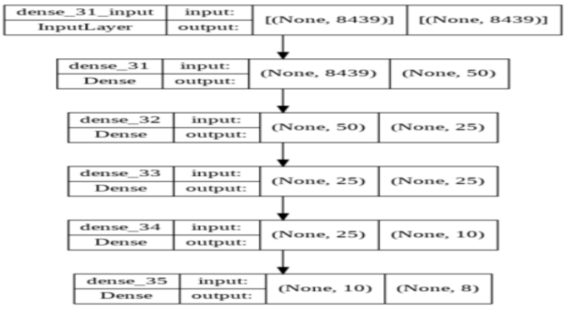


**Fig. 7:** Different Layers Details in the Proposed Model

In the first stage, the input is fed into the input layer. There are five dense layers in this proposed model. Three of the layers have ‘relu’ as an activation function. The fourth layer has ‘softmax’ as an activation function. We employed ‘categorical cross-entropy’ as the loss function and ‘adam’ optimiser to reduce the losses here. Figure 7 depicts these different layers in detail. Table 5 shows the Parameter and Values set for Associated Fine-tuned ANN and ML IoT Attack Predictor (Training)

**Table 5**: Parameter and Values set for Associated Fine-tuned ANN and ML IoT Attack Predictor (Training)

| Parameter | Value |
|---|---|
| activation | ‘relu’, ‘softmax’ |
| loss | ‘categorical_crossentropy’ |
| optimiser | ‘adam’ |
| metrics | ‘accuracy’ |
| epochs | 50 |
| verbose | 0 |

This model is saved in a file using Keras ‘plot_model’ which performs a dot format conversion on the model and saves the results to a file.

In the second stage, the model is loaded from the file to create associated ANN and ML systems. A new model is generated out of the results from the associated models. The testing dataset is fed into this new XGBoost classifier model to get better outcomes. The classifications are predicted using this model (Table 6).

**Table 6:** Parameter and Values set for XGboost IoT Attack Predictor for Testing

| Parameter | Value |
|---|---|
| random_seed | 43 |
| learning rate | 0.01 |
| n_estimators | 20 |
| n_jobs | -1 |
| max_depth | 10 |
| scoring | ‘f1’ |

## 6. Results and Discussion

Accuracy, precision, recall, F1-measure, and support are major indicators that are commonly employed and used in the procedure of assessing the success of the proposed methodology's efficiency. Each of these variables is calculated independently in the case of multi-class classification.

*Accuracy*

The capability of the system to correctly categorise the attack packet in accordance with the criteria outlined in table 1 is what is meant by the term ‘accuracy’. The mathematical form for accuracy is presented in the equation (2).

$$Acccuracy = \frac{TP + TN}{TP + TN + FN + FP} \qquad (2)$$

Table 7 presents the accuracy of different models experimented with in this study. The bar graph in figure 8 depicts the comparison of different model performance accuracies.

**Table 7**: The accuracy of different models in this study.

| Model | Accuracy |
|---|---|
| Random Forest Classifier | 68.97% |
| XGBoost Classifier | 82.54% |
| SVM Kernel Classifier | 82.36% |
| Associated Fine turned ANN & ML IoT Attack Predictor | 90.43% |

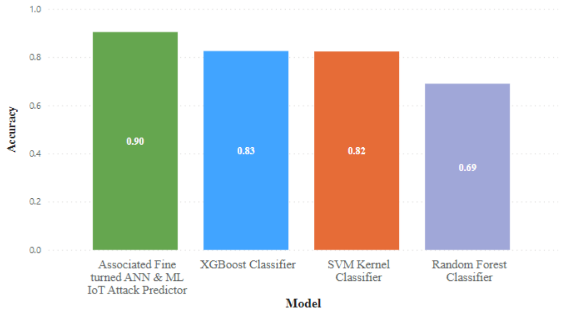


**Fig. 8:** Comparison of Model Accuracy

*Precision*

The proportion of precisely anticipated affirmative findings to the sum of all reported positive findings is what we mean when we talk about precision. A low percentage of false positives is associated with high precision. The mathematical form for precision is presented in the equation (3).

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (3)$$

**Table 8:** The precision of different models

| Code | Random Forest Classifier | XGBoost Classifier | SVM Kernel Classifier | Proposed IoT Attack Predictor |
|---|---|---|---|---|
| 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 1 | 0.5826 | 0.7605 | 0.8178 | 0.8646 |
| 2 | 0.6636 | 1.0000 | 1.0000 | 1.0000 |
| 3 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |
| 4 | 0.0000 | 0.3452 | 0.3411 | 0.6728 |
| 5 | 0.0000 | 0.3519 | 0.3356 | 0.6617 |
| 6 | 0.7835 | 0.9906 | 0.9991 | 1.0000 |
| 7 | 1.0000 | 0.9983 | 0.9931 | 0.9991 |

In the above table, *the* results of precision values are presented for different models we tested in this study. The table 8 shows that our proposed novel model increases precision in the case of classes 1, 4 and 5.

*Recall*

The term ‘recall’ refers to the proportion of accurately predicted positive data relative to the total number of the observations belonging to the ‘really positive’ class. The mathematical form for the recall is presented in the equation (4).

$$\operatorname{Re} call = \frac{TP}{TP + FN} \tag{4}$$

**Table 9:** The recall of different models

| Co de | Random Forest Classifier | XGBoos t Classifier | SVM Kernel Classifier | Propos ed IoT Attack Predictor |
|---|---|---|---|---|
| 0 | 0.8002 | 1.0000 | 1.0000 | 1.0000 |
| 1 | 0.9672 | 0.8368 | 0.7892 | 0.9102 |
| 2 | 0.9963 | 1.0000 | 1.0000 | 1.0000 |
| 3 | 0.9916 | 1.0000 | 1.0000 | 1.0000 |
| 4 | 0.0000 | 0.3217 | 0.3518 | 0.6186 |
| 5 | 0.0000 | 0.2586 | 0.3639 | 0.5998 |
| 6 | 0.4025 | 0.9981 | 0.9925 | 0.9991 |
| 7 | 0.3681 | 0.9913 | 0.9991 | 1.0000 |

Table 9 illustrates that the proposed IoT attack predictor predicts better recall values for classes 4 and 5. It also slightly increases the recall value for class 6.

*F1-score*

The F1 Score is determined by calculating the weighted average of the Precision and Recall categories. As a result, this metric considers the possibility of both false negatives and positives. The mathematical form for the F1-score is presented in the equation (5).

$$F1 - score = \frac{2(\operatorname{Re} call \times \Pr ecision)}{\operatorname{Re} call + \Pr ecision} \tag{5}$$

**Table 10:** The F1-score of different models

| Co de | Random Forest Classifier | XGBoos t Classifier | SVM Kernel Classifier | Propos ed IoT Attack Predictor |
|---|---|---|---|---|
| 0 | 0.8890 | 1.0000 | 1.0000 | 1.0000 |
| 1 | 0.7272 | 0.7968 | 0.8032 | 0.8868 |
| 2 | 0.7966 | 1.0000 | 1.0000 | 1.0000 |
| 3 | 0.9958 | 1.0000 | 1.0000 | 1.0000 |
| 4 | 0.0000 | 0.3330 | 0.3464 | 0.6446 |
| 5 | 0.0000 | 0.2981 | 0.3492 | 0.6292 |
| 6 | 0.5318 | 0.9944 | 0.9957 | 0.9995 |
| 7 | 0.5381 | 0.9948 | 0.9961 | 0.9996 |

Table 10 shows that our proposed IoT attack predictor increases F1-score of classes 1, 4, 5, and 6.

## 7. Conclusion

One of the emerging trends in the field of IT is pervasive computing. Thanks to technological advancements, people are increasingly foregoing bulky desktop computers in favour of lighter, more potent machines that can nevertheless handle complex computations and support a wide variety of wireless interfaces. Pervasive computing plays a crucial role because it enables the distribution of computer services to the physical spaces in which people perform their jobs, which raises questions of trustworthiness, confidentiality, and identification. The suggested study intends to construct a hybrid deep learning-based ubiquitous computing infrastructure to offer an optimization algorithm to these typical challenges. Deep neural network along with aXGBoost classifier is employed for the design of this proposed IoT attack predictor. We tested our selected input dataset on three other models for comparison of the model efficiency. Our proposed hybrid technique shows an accuracy of 90.43% which is far better than that of random forest, XGBoost, and SVM kernel classifiers. Upon further inspection it was found that the proposed model also increases precision, recall, and F1-score for 'Mirai-UDP Flooding', ‘Mirai-HTTp Flooding’, and ‘Mirai-Ackflooding’ attacks. The outcomes show the efficacy of the method and its capacity to compete with alternative approaches. The projected study has the potential to be broadened in the future to include the identification of discriminatory recommendation engines and its deployments on mobile devices, both of which would serve to verify the study's findings in authentic situations.

**Funding Information**

This research does not receive any specific grant from funding agencies in the public, commercial, or non-profit sectors.

## Data availability statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## References

[1] Z. Allam and Z. A. Dhunny, “On big data, artificial intelligence and smart cities,” *Cities, vol.* 89, pp. 80-91, 2019

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,’ *IEEE internet of things journal*, vol. 3, no. 5, pp. 637-646, 2016.

[3] C. Becker, C. Julien, P. Lalanda, and F. Zambonelli, “Pervasive computing middleware: current trends

and emerging challenges," *CCF Transactions on Pervasive Computing Interaction*, vol. 1, no. 1, pp. 10-23, 2019.

[4] J. Amarnath, P. G. Shah, and H. Chandramouli, "Advanced Key Management System (AKMS) for Security in Public Clouds," *in Evolution in Computational Intelligence: Springer*, pp. 573-582, 2021.

[5] F. J. I. P. C. Alt, "Pervasive Security and Privacy—A Brief Reflection on Challenges and Opportunities," vol. 20, no. 4, pp. 82-86, 2021.

[6] S. Kaushik and C. Gandhi, "Security and Privacy Issues in Fog/Edge/Pervasive Computing," *Fog, Edge, Pervasive Computing in Intelligent IoT Driven Applications*, pp. 369-387, 2020.

[7] A. Gouarir, G. Martínez-Arellano, G. Terrazas, P. Benardos, and S. Ratchev, "In-process tool wear prediction system based on machine learning techniques and force analysis," *Procedia CIRP*, vol. 77, pp. 501-504, 2018.

[8] D. Mukhametov, "Ubiquitous computing and distributed machine learning in smart cities, *in 2020 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF*), pp. 1-5, 2020. IEEE.

[9] X.-G. Zhang, G.-H. Yang, and S. Wasly, "Man-in-the-middle attack against cyber-physical systems under random access protocol," *Information Sciences*, vol. 576, pp. 708-724, 2021.

[10] V. Punitha, C. Mala, and N. Rajagopalan, "A novel deep learning model for detection of denial of service attacks in HTTP traffic over internet," *International Journal of Ad Hoc Ubiquitous Computing*, vol. 33, no. 4, pp. 240-256, 2020.

[11] Y. Liang, Y. Qin, Q. Li, X. Yan, L. Huangfu, S. Samtani, B. Guo, Z. Yu. "An Escalated Eavesdropping Attack on Mobile Devices via Low-Resolution Vibration Signals,*" IEEE Transactions on Dependable Secure Computing*, 2022.

[12] M. Salimitari, M. Chatterjee, and Y. P. J. I. o. T. Fallah,. "A survey on consensus methods in blockchain for resource-constrained IoT networks," *Internet of Things*, vol. 11, p. 100212, 2020.

[13] A. Waheed, A.I/ Umar, M. Zareei, N. Din, N.U. Amin, J. Iqbal, Y. Saeed, E.M. Mohamed, "Cryptanalysis and improvement of a proxy signcryption scheme in the standard computational model," *IEEE Access*, vol. 8, pp. 131188-131201, 2020.

[14] A. Mahfouz, A. Abuhussein, D. Venugopal, and S. Shiva, "Ensemble classifiers for network intrusion detection using a novel network attack dataset," *Future Internet*, vol. 12, no. 11, p. 180, 2020.

[15] X. Yao, F. Farha, R. Li, I. Psychoula, L. Chen, and H. Ning, "Security and privacy issues of physical objects in the IoT: Challenges and opportunities," *Digital Communications Networks*, vol. 7, no. 3, pp. 373-384, 2021.

[16] D. Y. Huang, N. Apthorpe, F. Li, G. Acar, and N. Feamster, "Iot inspector: Crowd sourcing labeled network traffic from smart home devices at scale," *Proceedings of the ACM on Interactive, Mobile, Wearable Ubiquitous Technologies*, vol. 4, no. 2, pp. 1-21, 2020.

[17] J. Shahid, R. Ahmad, A. K. Kiani, T. Ahmad, S. Saeed, and A. M. Almuhaideb, "Data protection and privacy of the internet of healthcare things (IoHTs)," *Applied Sciences*, vol. 12, no. 4, p. 1927, 2022.

[18] M. Langheinrich and F. Schaub, "Privacy in mobile and pervasive computing," *Synthesis Lectures on Mobile Pervasive Computing*, vol. 10, no. 1, pp. 1-139, 2018.

[19] H. A. Abdulghani, N. A. Nijdam, A. Collen, and D. Konstantas, "A study on security and privacy guidelines, countermeasures, threats: IoT data at rest perspective," *Symmetry*, vol. 11, no. 6, p. 774, 2019.

[20] G. Chen, M. Li, and D. Kotz, "Data-centric middleware for context-aware pervasive computing," *Pervasive mobile computing*, vol. 4, no. 2, pp. 216-253, 2008.

[21] E. Katsiri and A. Mycroft, "Linking temporal first-order logic with Bayesian networks for the simulation of pervasive computing systems," *Simulation Modelling Practice Theory*, vol. 19, no. 1, pp. 161-180, 2011.

[22] A. Padovitz, S. W. Loke, and A. Zaslavsky, "The ECORA framework: A hybrid architecture for context-oriented pervasive computing," *Pervasive mobile computing*, vol. 4, no. 2, pp. 182-215, 2008.

[23] S. I. Ahamed, H. Li, N. Talukder, M. Monjur, and C. S. Hasan, "Design and implementation of S-MARKS: A secure middleware for pervasive computing applications," *Journal of Systems Software*, vol. 82, no. 10, pp. 1657-1677, 2009.

[24] A. Boukerche and Y. Ren, "A trust-based security system for ubiquitous and pervasive computing environments," *Computer communications*, vol. 31, no. 18, pp. 4343-4351, 2008.

[25] P. Yu, X. Ma, J. Cao, J. J. P. Lu, and M. “Computing, Application mobility in pervasive computing: A survey,” vol. 9, no. 1, pp. 2-17, 2013.

[26] A. B. Usman and J. Gutierrez, “Toward trust based protocols in a pervasive and mobile computing environment: A survey,” *Ad Hoc Networks*, vol. 81, pp. 143-159, 2018.

[27] G. Carullo, A. Castiglione, G. Cattaneo, A. De Santis, U. Fiore, and F. Palmieri, “Feeltrust: providing trustworthy communications in ubiquitous mobile environment,” *in 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA),* pp. 1113-1120, 2013. IEEE.

[28] M. K. Denko, T. Sun, and I. Woungang, “Trust management in ubiquitous computing: A Bayesian approach,” *Computer Communications*, vol. 34, no. 3, pp. 398-406, 2011.

[29] M. Ivanova, “Researching affective computing techniques for intelligent tutoring systems,” *in 2013 International Conference on Interactive Collaborative Learning (ICL)*, pp. 596-602, 2013. *IEEE*.

[30] Y. Chang, W. Li, and Z. Yang, “Network intrusion detection based on random forest and support vector machine,” *in 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC)*, vol. 1, pp. 635-638, 2017.

[31] U. Tariq, T. A. Ahanger, M. Nusir, and A. Ibrahim, “A Pervasive Computational Intelligence based Cognitive Security Co-design Framework for Hype-connected Embedded Industrial IoT,” *International Journal of Computers, Communications Control*, vol. 16, no. 2, 2021.

[32] I. Ullah and Q. H. Mahmoud,.A scheme for generating a dataset for anomalous activity detection in iot networks, *in Canadian Conference on Artificial Intelligence*, pp. 508-520, 2020.

[33] K. S. Athrey, Tutorial on Keras, *ed: Cap*, 2018.