Adversarial Defense: A GAN-IF Based Cyber-security Model for Intrusion Detection in Software Piracy

U. Kumaran[1*], S. Thangam[2], T.V. Nidhin Prabhakar[3], Jana Selvaganesan[4] and H.N. Vishwas[5]

[1*] Assistant Professor (SG), Department of Computer Science and Engineering Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India. u_kumaran@blr.amrita.edu, Orcid: https://orcid.org/0000-0002-0160-2703

[2] Assistant Professor (SG), Department of Computer Science and Engineering Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India. s_thangam@blr.amrita.edu, Orcid: https://orcid.org/0000-0003-2251-3651

[3] Assistant Professor, Department of Computer Science and Engineering Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India. tv_nidhin@blr.amrita.edu, Orcid: https://orcid.org/0000-0002-5303-6918

[4] Professor, Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India. drsjana@veltech.edu.in, Orcid: https://orcid.org/0000-0001-7829-1301

[5] Assistant Professor (SG), Department of Computer Science and Engineering Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India. hn_vishwas@blr.amrita.edu, Orcid: https://orcid.org/0000-0002-9585-4097

Received: August 12, 2023; Accepted: October 18, 2023; Published: December 30, 2023

Abstract

Software-piracy continues to be most critical distress, posing grave threats to digital-assets and financial stability. Traditional Intrusion Detection systems (IDS) often battles hard to identify latest piracy attempts owing to their dependence on pre-established patterns. To effectively address this we attempt to suggest innovative approach leveraging DL based Generative Adversarial Networks (GANs) and ML based Isolation Forest (IF) for detecting software piracies. Our proposed GAN-IF based cyber-security model performs its functions by training a Generator network to mimic the behavior of genuine software applications. Discriminator network discriminates between legitimate and pirated software. Isolation Forests assists in detecting anomalies in diverse conditions, including unseen attacks. Integrated training based on DL and ML framework enables efficient learning and adaptation with respect to piracy challenges, making it highly-successful against prior known threats. There are several DL models which are utilized in IDS operations having limitations in terms of robustness, interpretability. Utilizing GAN in the context of cyber-security to combat software-piracy can have noteworthy merits since GANs can precisely identify forged software as they are skilled at generating fake content resembling actual. Training a GAN on legitimate software, helps to learn and identify disparities in pirated versions. Isolation-Forest can detect anomalies in software distribution networks or user behavior with respect to software usage by recognizing abnormal patterns indicating software piracy, like illegal access or sharing of software licenses. Our proposed

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 14, number: 4 (December), pp. 96-114. DOI: *10.58346/JOWUA.2023.I4.008*

*Corresponding author: Assistant Professor (SG), Department of Computer Science and Engineering Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India.

model combines GANs and Isolation Forests, excels at accurately detecting subtle indicators of software piracy, a capability that traditional methods may fail to recognize. ML-DL integrated model continuously learns and updates its detection capabilities in response to evolving piracy tactics, making it resilient against zero-day attacks, polymorphic malware. Through adversarial training, ml-model minimizes false alarms and focuses only on genuine threats. In our evaluation, we demonstrate the effectiveness of GAN-IF based cyber-security model in detecting software piracy attempts across various scenarios. Results indicate that our approach outperforms traditional solutions in terms of detection accuracy and adaptability.

Keywords: IP, IDS, GAN, IF, Zero Day Attacks, RNN, LSTM, CNN, Auto-encoders, Adversarial Networks.

1 Introduction

Cyber-security (Aldhyani, T.H., 2023) is a significant landscape in modern digital era, as businesses and individuals around the globe more and more rely on computers, the internet, and other digital advancements to conduct their everyday dealings, communicate, and stock up confidential information. With this never before increased dependence on latest technologies and due to excessive proliferation of IoT devices, the apparent risk of cyber threats has risen considerably (Diro, A.A., 2018). These threats could present themselves in multiple forms, like viruses, malware, ransom ware, phishing attacks, data breaches, and so on. Traditional security measures like firewalls, encryption mechanisms and anti-virus software tools were not competent enough to deal with modern hacking strategies thus paving way for AI, Machine learning and Deep learning techniques (Al-Garadi, M.A., 2020) (Nagarajan, S.M., 2022) that have become invaluable tools in various domains including the field of cyber-security (Figure 1). These approaches contribute noteworthy solutions to safeguard the system against evolving threats through anomaly detection, where AI models can learn the characteristics of a normal network or user behavior patterns and immediately alerts when it detects any unusual deviation from normalcy which may possibly indicate a cyber-attack (Chen, D., 2021). Furthermore, ML models can augment the security analysis of network traffic through User authentication and access control and intrusion detection (Otoum, Y., 2022) (Otoum, S., 2019) by flagging potentially destructive network activity in real-time and helps to prevent unauthorized access even if valid credentials are compromised (Aldriwish, K., 2021).

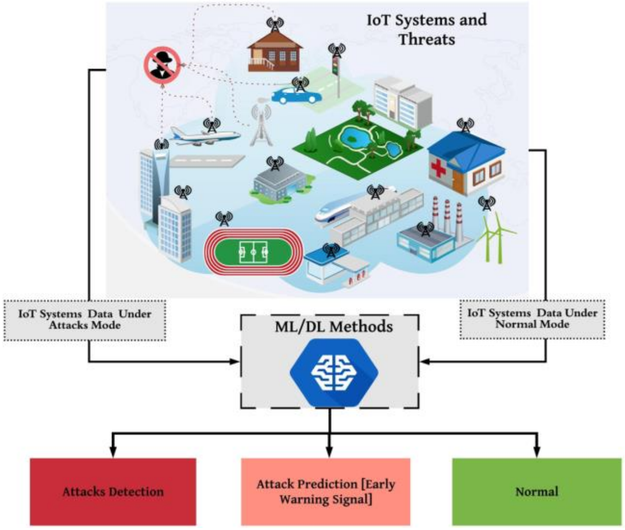

IoT Systems and Threats
IoT Systems Data Under Attacks Mode
ML/DL Methods
IoT Systems Data Under Normal Mode
Attacks Detection
Attack Prediction [Early Warning Signal]
Normal

Figure 1: Role of IoT, ML and DL in Threat Detection (Al-Garadi et al,2018)

In addition to ML models like Support Vector Machine(SVM),random Forest(RF),Decision Tree(DT),Isolation Forest(IF),Naïve Bayes (NB) etc., the recent years has been witnessing the surge of deep learning models (Sultana, N., 2019) which offers effective facial recognition, biometric authentication systems, malware analysis and user authentication through several techniques, such as convolutional neural networks (CNNs),recurrent Neural Networks(RNNs),Generative Adversarial Networks(GANs), Deep Belief Networks (DBNs),Long Short Term Memory(LSTM) etc which can excel in analyzing humongous datasets to identify patterns and attributes in malicious code, aiding in speedy threat detection and categorization (Dina, A.S., 2021).

– Legal Consequences
– Financial Consequences
– Security Risks
– Negative Impact on Economy
– Ethical and Reputational Consequences

Figure 2: Software Piracy and Its Effects

Software piracy which is a part of cyber security indicates illegal copying, distribution, or utilization of computer software without the appropriate authorization from the legal copyright holder or software developer (Gümüşbaş, D., 2020). It includes unlawful duplication, installation, or sharing of software applications that are guarded by copyright laws. Some software pirates employ key generators or software cracks to evade activation / registration processes by generating false license keys, while cracks can alter the software to immobilize licensing checks (Verma, A., 2020). Software piracy is illegal in most countries and can have severe consequences, including legal actions, fines, and imprisonment (Kim, J., 2020). Moreover, while using pirated software versions, there are possibilities of exposing the users and enterprises to innumerable security risks, as fake software can contain malware or present as a backdoor for further threats. Software piracy has a major financial impact on software developers along with technological industry as a whole that may result in missing revenues, hampers innovation, and would increase the software prices for rightful clients and purchasers (Figure 2). Several challenges are faced in combating software piracy which is complex and multifaceted like digital distribution, anonymity, difficulty in licensing, user resistance etc often arise due to the transforming scenery of technology, wide reach of internet which serve as the possible motivations behind this study (Piplai, A., 2020) (Shankar, S., 2019).

Tackling software piracy using deep learning offers advanced solutions to detect, prevent, and mitigate unauthorized distribution and usage of pirated software. Deep learning models can be trained for behavior analysis, content matching, anomaly detection, chatbot based reporting, digital finger printing etc which can assist in gathering and analyzing threat intelligence related to software piracy, thereby facilitating various organizations to stay informed about rising piracy trends and tactics (Tian, Z., 2019). Our proposed GAN-IF based anomaly detection model can improve the accuracy of identifying malicious software usage patterns, leading to more effective threat detection. Generative Adversarial Network (GAN) is a type of artificial neural network architecture which is designed for generative tasks and has gained considerable interest in the recent times due to their ability to generate realistic and high-quality data samples through their generator and distributor components (Benaddi, H., 2022). GANs can be used for anomaly detection in software piracy by leveraging their ability to model normal data distributions and recognize divergences from those distributions. Isolation Forest (IF) operates by segmenting the dataset into subsets using random splits along feature dimensions and builds a tree-like structure where anomalies are expected to be isolated in a shorter number of splits compared to normal data points (AbuAlghanam, O., 2023). This attribute makes it highly effective for detecting outliers in high-dimensional datasets. Real-time monitoring of software usage and network traffic through the suggested GAN-IF cyber security framework enables rapid detection and response to anomalies as they occur (Pinto, L., 2022). Following are some of the major contributions of our suggested model:

- Our proposed model can adapt itself to evolving software piracy schemes and rising threats by incessantly learning from new data.
- Through concurrent analysis of multiple data modalities, like network traffic, user behavior, and content matching, a broad perspective of potential anomalies can be accomplished.
- Reduced False Positives and Enhanced Security stance by generating alerts and initiate response actions swiftly, thereby minimizing the time and Security vulnerabilities.
- By effective detection and prevention of software piracy, the integrated GAN-IF model contributes towards cost savings, enhanced user experience and legal compliance-based IP protection.

These contributions are aimed at enhancing cyber-security endeavors and securing intellectual property protection in the context of software piracy and also help organizations to effectively alleviate the risks connected with illegal software distribution and usage while preserving a safe and acquiescent software environment. Organization of our proposed study is as follows: Section 1 presents introductory information on the cyber security background along with challenges addressed using ML and DL integration has been explained followed by Section 2 which discusses the relevant research and prior works related to proposed topic. Section 3 elaborately explains the proposed GAN-IF based intrusion detection methodology. Section 4 presents the empirical findings and results in a clear and organized manner. Ultimately Section 5 concludes the study by summarizing the key findings and their significance.

2 Related Works

Recent advancements and the emerging interdisciplinary interactions between cyber security and artificial intelligence such as machine learning and deep learning, enables construction of smart models to carry out malware classification and outlier detection through threat intelligence sensing based on dataset-based learning, and decision making (Ahmad, Z., 2021). Thus, AI models offer cyber security defense and fortification to combat the challenges posed by software piracy and malware attacks. This section attempts to present prominent research works conducted by various scholars whose models are based on ML and deep learning technologies which helps to detect potential cyber security related threats across the IoT network.

Sharma et al (2023) suggested a novel IDS system for IoT networks which employs filter-based feature selection through Deep Neural Network (DNN).Our proposed model extracts highly correlated features and classifies the threats based on classification scores. DL model has been fine-tuned with various hyper parameters to train using UNSW-NB15 dataset that contains various attack classes and accuracy has been reported as 97%. Aldhyani et al (2023) emphasized on the rising number of attackers who are progressively targeting the industrial devices which are connect to the network. A network intrusion detection system has been deployed to offer an effective and compliant intrusion detection system using CNN–LSTM has been proposed for detecting DDoS attacks. CIC-DDoS2019 dataset was used to detect different types of attacks, several network traffic datasets, were used to test the performance of proposed approaches with a score of 93% in accuracy (Sharma, B., 2023).

Saheed et al (2022) focuses on applying ML for IDS in IoT environments. Feature extraction was done through application of min–max principle which restricts the information leakage (Saheed, Y.K., 2022). Dataset deployed is a mixture of existing attacks and regular activities which forms the simulated network traffic setting to detect different attack types. Dimensionality reduction using Principal Component Analysis (PCA) is performed followed by utilization of 6 different machine learning models for classification. Performance analysis and experimental results in terms of accuracy, recall, AOC-ROC, and other findings present an accuracy of 96.9%. Aldriwish et al (2021) came up with a malware model based on Deep Convolutional Neural Networks (DCNNs) which along with TensorFlow Deep Neural Networks (TFDNNs) helps in detecting software piracy threats in accordance with source code bootlegging. Based on the analysis conducted on Google Code Jam dataset, classification performance and accuracy has been measured to be around 95% (Aldriwish, K., 2021).

Gümüşbaş et al (2021) provided a systematic review on recent approaches with respect to IDS in terms of various methods, performance outcomes, and constraints along with benchmark datasets for cyber-security. This evaluation is intended to offer a road map for software developers and research personnel who would like to comprehend the potential DL methods and datasets that were used to train these models (Gümüşbaş, D., 2020). Grover et al (2020) presented a detailed analysis on various threat detection frameworks by investigating several issues involved in mitigating malware and software piracy attacks. ANN and CNN based implementation has been suggested to capture, classify and predict the threats which marked a performance score of 95.4% (Grover, M., 2020).

Tian et al (2019) delved into the limitations encountered by conventional IDS and firewalls as they struggle with novel network environments and the solutions offered by modern technologies like IoT, machine learning and deep learning which are rapidly emerging to be the indispensable tools in almost every domain. A web attack recognition framework has been designed to detect threats on cloud edge devices (Tian, Z., 2019) (Kumaran, U., 2021). Several synchronized DL models were used to augment the permanence of the system whose performance was compared with existing systems by means of varied datasets. Experimental results in terms of detection rate have been around 97.31% which demonstrates the system's competence in terms of threat detection. Diro et al (2018) utilized DL for threat recognition in cyberspace to come up with a robust approach to deal with minor alterations or fresh attacks due to the complex high-level feature extraction ability possessed by this latest technology. Self-learning architecture of DL helps to discover hidden patterns from training data so that it becomes possible to discriminate them from regular network traffic (Diro, A.A., 2018). By comparing the proposed RNN based approach with traditional ML techniques, evaluation of the model has been done to ensure that centralized detection systems using deep learning model are more effective in threat management and discovery whose detection rate is around 98%.

Van et al (2017) structured their anomaly-based intrusion detection technique around the principle of arriving at an optimal solution to deal with the evolving nature of modern attacks and challenges in terms of adapting to varying dynamic network environments, insufficient availability of labeled data and increased false positive rates. Auto encoder-based IDS has been proposed which show causes their sensitivity in arriving at superior quality classification, their capability to realize a portion of their knowledge from partial data along with their adaptability (Van, N.T., 2017). KDDCup99 network traffic connections have been utilized to simulate the effectiveness of suggested system in detecting outliers and categorize invasions into 5 different classes with the accuracy reported as 94% in accordance with network data sources.

Research Gaps Identified and Potential Solutions

As far as the realm of cyber security is concerned, due to the evolving characteristic of digital environments, further investigation and innovation are always in demand to reinforce the digital defenses. One major gap lies in the uncovering and alleviation of zero-day vulnerabilities, which are unanticipated security flaws that can be taken as an advantage before executing any patches or setting up any potential safeguards. Even the most efficient and precise method may fail to uncover and thwart these threats. Furthermore, as AI and ML are increasingly getting integrated into security systems, future research must focus on augmenting these technologies' resilience against adversarial attacks, where attackers may manipulate data inputs to mislead these models. Security of IoT devices, their resource-constrained and ubiquitous nature, too poses another challenge that requires specialized research along with the ability to deal with quantum computing whose fast emergence raises concerns regarding existing cryptographic methods. User-centric security practices, privacy-preserving approaches, behavioral analysis techniques are few research gaps that represent prospective potentials to fortify cyber-security space and demands securing digital systems by meeting the challenges posed by ever-evolving threat landscape.

3 Proposed GAN-IF Cyber Security Model

Recommended GAN-IF-based Intrusion Detection System (IDS) for software piracy detection involves a systematic series of phases. Initially, a simulated dataset emulating the legitimate software usage and example instances of software piracy is collected and meticulously prepared. Data preprocessing including normalization and feature engineering is carried out, to enhance the dataset's suitability for analysis. Subsequently, a Generative Adversarial Network (GAN) model is trained to generate synthetic data closely resembling authentic software usage patterns (Seo, E., 2018). This artificial data, along with genuine data, is then subjected to an Isolation Forest (IF) algorithm for anomaly detection. IF algorithm identifies deviations from expected behavior, potentially flagging instances of software piracy. Following are the significant phases involved in designing the proposed GAN-IF based cyber-security system:

- Data collection and pre-processing
- GAN model based Synthetic data generation
- Isolation Forest based Anomaly Detection

Data Collection and Pre-processing

Data collection and preprocessing phase in our software piracy detection framework using GAN and IF is a decisive step since it lays the foundation for subsequent stages of intrusion detection framework. This phase involves the collection of relevant data related to software usage, user behaviors, network traffic, or system logs, and then preparing this data for further analysis and model training. Proposed work has a simulated dataset for piracy detection, as it combines the techniques like GAN (Generative Adversarial Network) and Isolation Forest (IF), offers several advantages over relying solely on public datasets. Using an indigenous dataset for a GAN-based piracy detection model presents several distinct advantages as it offers a level of regional relevance that can be crucial for accurately identifying piracy practices specific to certain communities which can further enhance the model's precision in detecting those particular forms of piracy. Once the data is collected, it undergoes preprocessing to make it appropriate for further analysis and training within the GAN-IF-based IDS framework. Data pre-processing phase involves identification and handling of missing values, and other inconsistencies in the dataset. Cleaning ensures that data is of utmost quality and devoid of any errors that could bring in bias into the proposed model. Principal component Analysis (PCA) and Normalization ensures that relevant features are extracted and they are on similar scales which are vital for Isolation Forest, which rely on distance metrics, as well as for neural networks in the GAN. Preprocessed dataset is then split into training, validation, and test sets in an 80-10-10 ratio. Training set is used to train the GAN and Isolation Forest, while the validation and test sets serve for model evaluation. Categorical data, such as software application names and user roles, is label-encoded and other personally identifiable information is anonymized, to safeguard confidential data and to protect user privacy and ensure agreement with data safety regulations.

GAN Model based Synthetic Data Generation

A Generative Adversarial Network (GAN) consists of two neural networks, generator and discriminator, which works in opposition to each other (Figure 3). Generator learns to generate data samples that closely resemble the real data. It takes random noise as input and transforms it into data samples. Objective of this module is to generate synthetic data which is completely impossible to differentiate from genuine data. Discriminator is another component of GAN that evaluates whether a particular data sample is authentic from the training dataset or forged by the generator. It takes both actual and artificial data samples as input and generates a probability score demonstrating the likelihood that input data is genuine. Thus, the discriminator's goal is to properly categorize true and fake data.

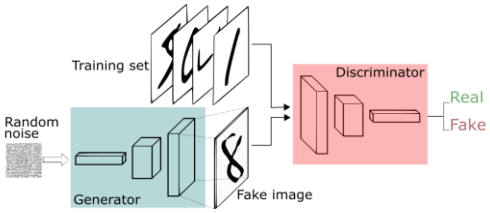


Figure 3: GAN Architecture

Generator and discriminator are trained using different loss function while the generator's loss function persuades it to produce data that tricks the discriminator, implying the fact that loss is minimized when discriminator cannot distinguish real from fake. Discriminator's loss function determines its capability to accurately classify data as original or artificial. GAN training procedure involves a game theory-based process where the generator and discriminator compete with each other. Training process of GAN can be mathematically represented as an optimization problem involving generator (G) and the discriminator (D). Our objective of training is to determine the parameters for both networks that results in Nash equilibrium, where data generated by generator is impossible to tell apart from original data, and the discriminator could not discriminate between authentic and forged data. Generator maps a random noise vector z to generate data samples $G_{en}(z)$ which can be represented as:

$$G_{en}: z \rightarrow G_{en}(z) \quad (1)$$

Generator is represented as a neural network 'G_{en}' that maps 'z' to 'G(z)'

$$G_{en}(z) = G_{en}(z; \theta\_G_{en}) \quad (2)$$

Where θ_G_{en} represents the probability distribution parameters of generator network and Gen(z; θ_Gen) denotes generator's output which is the synthetic data generated from random noise z.

Discriminator is represented as a neural network 'D_{isc}' that maps 'x' to a probability score indicating if a given data sample is authentic (1) or fake (0) which can be represented as:

$$D_{isc}: x \rightarrow D_{isc}(x), \text{ where } D_{isc}(x) \in [0, 1] \quad (3)$$

$D_{isc}(x) = D_{isc}(x; \theta\_D_{isc})$, where θ_D_{isc} represents the parameters of the discriminator network. Discriminator's aim is to increase the probability of accurately classifying genuine data as original and bogus data as forged one which is a binary classification problem. Loss function for the discriminator can be represented as:

$$L\_{}_{Gen}(\theta\_G_{en}) = E[\log(1 - D_{isc}(G_{en}(z; \theta\_G_{en}); \theta\_D_{isc}))] \qquad (4)$$

where $L_{Gen}$is generator's loss and $D_{isc}(G_{en}(z))$ is the probability assigned by the discriminator to generated data.Negative sign persuades the generator to make the most of the above function, which indicates it needs to generate data that is more likely to be categorized as real by the discriminator.

Generator's objective is to minimize this loss function, thereby effectively making the generated data look more convincing and difficult for the discriminator to tell apart from real data.

$$\text{Maximize } E_x{\sim}P_{data}(x)[\log(D_{isc}(x))] + Ez{\sim}P_z(z)[\log(1 - D_{isc}(G_{en}(z)))] \qquad (5)$$

Proposed discriminator's loss can be denoted as

$$L\_{}_{Disc}(\theta\_{}_{Disc}) = E[\log(D_{isc}(x; \theta\_D_{isc}))] + E[\log(1 - D_{isc}(G_{en}(z; \theta\_G_{en}); \theta\_D_{isc}))] \qquad (6)$$

$$\text{Minimize } E_z{\sim}P_z(z)[\log(1 - D_{isc}(G_{en}(z; \theta\_G_{en})))] \qquad (7)$$

Generator aims to minimize this loss by adjusting its parameters $\theta\_{}_{Gen}$ to generate more convincing data.

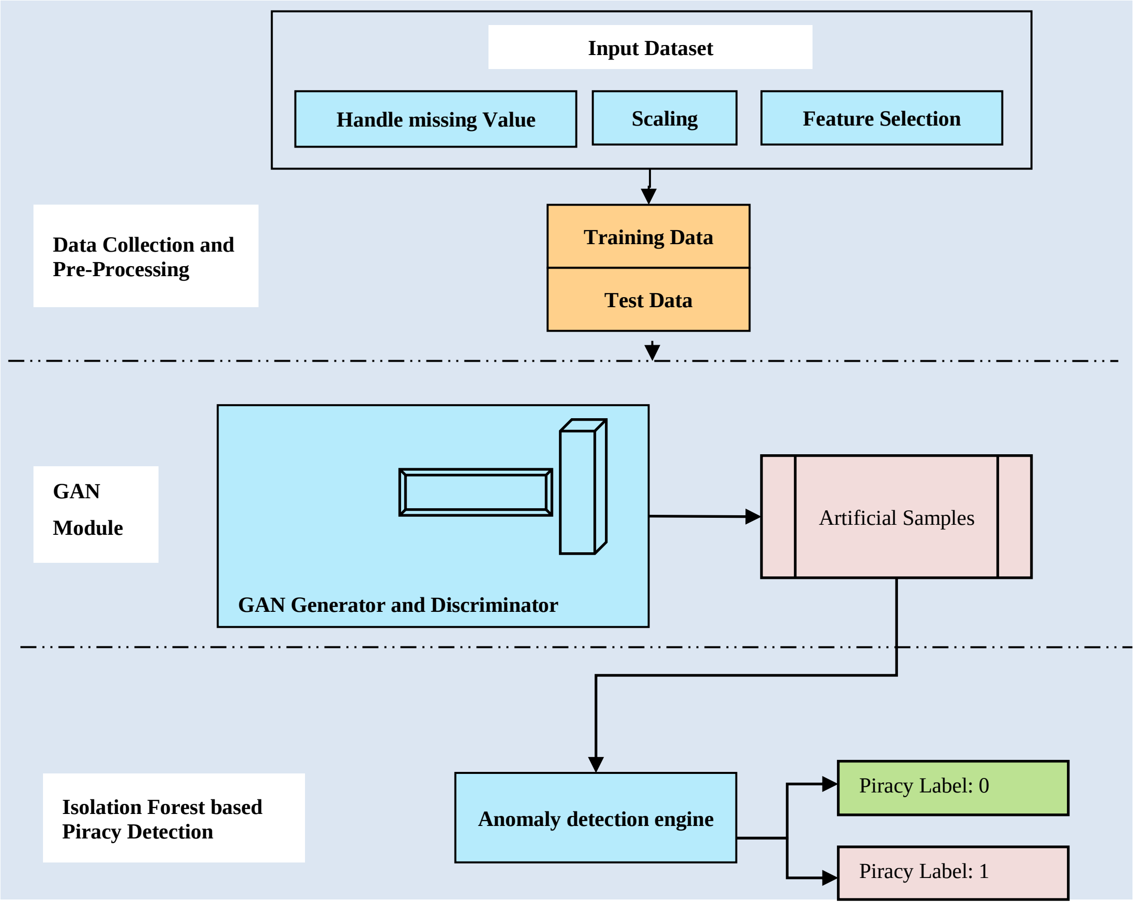


Figure 4: Proposed Model Architecture

GAN training procedure involves alternating between updating the discriminator and updating the generator iteratively where discriminator's parameters θ_D are updated to maximize its loss, while the generator's parameters θ_G are updated to minimize its loss. This adversarial process continues till convergence when generator produces high-quality synthetic data. GAN training makes use of gradient-based optimization techniques like Adam. Generator's loss and discriminator's loss is calculated as the sum of two binary cross-entropy terms for real data and fake data. Overall loss for the GAN is the total of the generator and discriminator losses. Distinct attribute of GAN architecture is its adversarial training where generator and discriminator learn from each other's mistakes, eventually leading to production of realistic artificial data. Our proposed model architecture is presented in Figure 4.

Isolation Forest based Anomaly Detection

After training GAN and generate synthetic data, the original data and the fake data are fed as an input to the Isolation Forest (IF) component. IF works on feature space and identifies anomalies based on variations from learned feature patterns (Khan, M.A., 2021). Anomaly scores from real and artificial data are used to detect abnormal patterns. IF is an unsupervised ML algorithm that isolates anomalies in a dataset by measuring the average number of splits to segregate an instance in a random forest of isolation trees using a tree shaped structure. Let (H(x)) denotes the height of a data point x in an isolation tree which denotes the number of edges traversed from root to data point. IF partitions the feature space repeatedly until each data point is separated in its own leaf node. Path length (PL(H(x))) for a data point x is calculated as:

$$[PL\ (H(x)) = C(H(x))] \qquad (8)$$

Where (C (H)) is a constant that depends upon number of data points (n) used to build the tree. In the case of binary trees,

$$C(H)=2(1-(\ H-1)/n) \qquad (9)$$

For a forest of T number of isolation trees, the expected path length (E_{PL}(H(x))) for a data point (x) in the forest is calculated as:

$$E_{PL}(H(x))=T\sum^{T}_{t-1}TE_{PLt}(H(x)) \qquad (10)$$

Anomaly score (AS) for a data point x is calculated based on its expected path length:

$$AS(x,n)=2^{-E(h(x))/c(n)} \qquad (11)$$

Where (c(n) is a constant that depends on the average path length for n data points, calculated as

$$C(n)=2H(n-1)-[2(n-1)/n] \qquad (12)$$

A predefined threshold is utilized to classify data points as anomalies or normal based on their anomaly scores. Data points whose anomaly scores greater than the threshold are considered anomalies. Isolation Forest module thus isolates outliers by comparing them to normal data points, leading to lesser average path lengths and maximum anomaly scores. This makes it a competent approach for detecting anomalies, particularly in high-dimensional datasets. Pseudo code for the proposed GAN-IF model is presented below.

GAN-IF Model Pseudo Code

```
 Step 1: Data Preprocessing
data = load_and_preprocess_data()
Step 2: Feature Engineering to identify and select relevant features that may indicate piracy
features = select_features(data)
Step 3: Initialization and Training the GAN to generate synthetic data
GAN = initialize_GAN()
GAN.train(data)
```

```
 Step 4: Generate Synthetic Data using the trained GAN
Artificial_data = GAN.generate_samples()
Step 5: Combine Real and artificial Data to create an augmented dataset
augmented_data = concatenate(data, artificial_data)
 Step 6: Training the Isolation Forest
 Initialize and train the Isolation Forest model
Isolation_forest = initialize_isolation_forest()
isolation_forest.train(augmented_data)
Step 7: Anomaly Detection using the trained Isolation Forest
Anomaly_scores = Isolation_forest.detect_anomalies(data)
Step 8: Thresholding to classify instances as normal or anomalous
Threshold = determine_threshold(anomaly_scores)
Step 9: Reporting and Action by Flagging instances with anomaly scores above the threshold as
potential piracy incidents
for instance in data:
    if anomaly_scores[instance] > threshold:
report_piracy_incident(instance)
```

4 Outcome of Proposed Approach and Related Summary

Analyzing and discussing the results of a cyber security piracy detection model based on Generative Adversarial Networks (GANs) and Isolation Forests is a critical step in comprehending its performance and potential for real time applications. We choose DCGAN (Deep Convolutional GAN) architecture due to its capability to generate complex data, such as network traffic patterns. A diverse indigenous dataset comprising legitimate software samples and known pirated software samples has been utilized whose features include File size, version, file extensions, code obfuscation presence, and the presence of specific keywords. Binary labels indicate 0 for legitimate, 1 for pirated. The size of dataset is 10,000 software samples (7,000 legitimate, 3,000 pirated). A DCGAN generator generates synthetic software samples that resemble legitimate software while its discriminator distinguishes between actual and fake software samples.

Scikit- learn's Isolation Forest (Tao, X., 2018) identifies anomalies in the dataset that do not conform to the characteristics of either legal or recognized pirated software. Python 3 has been used for simulation with Scikit-learn, PyTorch for GAN development and additional Libraries like Pandas, NumPy has been used for data manipulation and visualization. Jupyter Notebook has been used for experimentation. Hardware Requirements includes multi-core processor (Intel Core i7) for data preprocessing and model training while a CUDA-compatible GPU (NVIDIA GeForce) is used for training DCGAN, to accelerate its training process with16 GB RAM for handling datasets and for carrying out deep learning tasks efficiently along with sufficient storage capacity for storing the dataset, models, and in-between results.

Experimental Workflow

Data Preprocessing: Involves loading and preprocessing the dataset, including feature extraction and label encoding.

GAN Training: DCGAN is trained using PyTorch, generator learns to generate fake software samples, while the discriminator learns to differentiate between actual and synthetic samples.

Isolation Forest Training: Scikit-learn based training is done by combining real legal and synthetically generated legitimate samples from GAN, and prior known pirated samples. When a new software sample is encountered; it is passed through the GAN to generate a synthetic version followed by Isolation Forest based assessment which identifies if particular software is real or fake by carrying out anomaly detection operation.

Reporting: If a software sample is flagged as an anomaly, it indicates a potential software piracy. System logs the result and generates alert-based reports.

Simulation parameters for both DCGAN and If are presented in Tables 1 and 2.

Table 1: Simulation Parameters for DCGAN

Hyper Parameters	Values
Learning rate	0.0002
Batch Size	32
Noise Dimension	100
Generator layers	Conv2DTranspose, ReLU
Discriminator layers	Conv2D, LeakyReLU
Optimizer	Adam [beta1=0.5, beta2=0.999]
Training Epochs	1000
Loss function	Binary Cross-Entropy

Table 2: Isolation Forest Hyper-parameters

Simulation Attributes	Values
Number of Trees	100
Maximum samples	256
Contamination	0.004
Maximum Features	Auto
Bootstrap	false
Random State	32
CPU jobs	1

Performance Metrics based Evaluation

Evaluation metrics, such as precision, F1-Score, FPR and recall, are employed to assess the GAN-F model's performance. Fine-tuning and iterative refinement are carried out to enhance detection accuracy, and finally by deploying in a production environment for continuous monitoring, timely detection and prevention of software piracy attempts is made.

Total Test Instances: 10,000
True Positives (Correctly Detected Piracy): 950
False Positives (Legitimate Users Incorrectly Flagged as Piracy): 50
False Negatives (Actual Piracy Missed by the System): 75
True Negatives (Correctly Identified Legitimate Users): 8,925
Estimation of performance metrics are as follows:

Accuracy is a metric that quantifies the proportion of correctly predicted instances (both true positives and true negatives) out of the total number of instances in a dataset. **Accuracy** = (TP + TN) / (TP + TN + FP + FN)

= (950+8925)/(950+8925+50+75)=0.9875

Precision: Precision is the ratio of correctly detected piracy instances to the total instances flagged as piracy by the system.

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

Precision = 950 / (950 + 50) = 0.95

Precision is 95% of the instances flagged as piracy by the system were actual cases of piracy.

Recall: Recall is the ratio of correctly detected piracy instances to all actual instances of piracy.

Recall = True Positives / (True Positives + False Negatives)

Recall = 950 / (950 + 75) = 0.92

Recall also known as the TPR, is 0.92, meaning that the system successfully detected 92% of all actual instances of piracy.

F1-Score: F1-score is a combination of precision and recall, providing a single metric that balances both.

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

F1-Score = 2 * (0.95 * 0.92) / (0.95 + 0.92) = 0.884

F1-score is 0.884, which represents a balance between precision and recall.

False Positive Rate (FPR): FPR is the ratio of falsely identified legitimate users as pirates to all actual legitimate users.

FPR = False Positives / (False Positives + True Negatives)

FPR = 50 / (50 + 8,925) = 0.005

FPR is 0.005, indicating that 0.5% of legitimate users was incorrectly flagged as pirates.

Empirical Findings

In this software piracy-based anomaly detection system, following features related to software usage and behavior are used to build the dataset for training and testing. These features help the system to differentiate between legitimate software practice and potential piracy. File-Related Feature like name, size, modification date, access frequency and access timestamps. User Behavior Features like User ID, Session duration, Average session length, Number of concurrent sessions, Frequency of software usage and Abnormal login patterns. Network Activity Features like IP address, MAC address, Network traffic volume, Number of unique IP addresses, Geographic location based on IP address, Network ports used and unusual network activity like excessive data transfer. Software Version Information, License key or activation status and Software updates or patching history.

Hardware-Related Features like machine identifier, specifications and Number of hardware changes / replacements. User Profile Features like role / permissions, department, access history, authentication methods. Behavioral Pattern based features like Deviation from normal behavior, usage hours. License related features like type, expiration date and activation history. Table 3 presents the dataset features and event logs.

Table 3: GAN-IF Model Dataset Features

User ID	File Name	File Size (KB)	Access Count	User Behavior Score	Network Traffic (MB)	License Status	Region	Piracy Label
1	document.doc	150	25	0.8	2.5	Licensed	US	0
2	spreadsheet.xls	300	15	0.7	1.8	Trial	Canada	0
3	presentation.ppt	200	10	0.6	2.2	Expired	UK	1
4	report.doc	100	30	0.9	1.5	Licensed	US	0
5	budget.xlsx	250	20	0.7	1.9	Licensed	Canada	0
6	analysis.xls	280	18	0.75	2	Trial	Germany	0
7	proposal.doc	180	22	0.85	2.1	Licensed	US	0
8	presentation.ppt	210	12	0.65	2.3	Expired	Canada	1
9	sales_report.xls	320	28	0.92	1.7	Licensed	UK	0
10	data.xlsx	270	16	0.68	2.4	Trial	Germany	0

Table 4: User Behavior based Outcomes

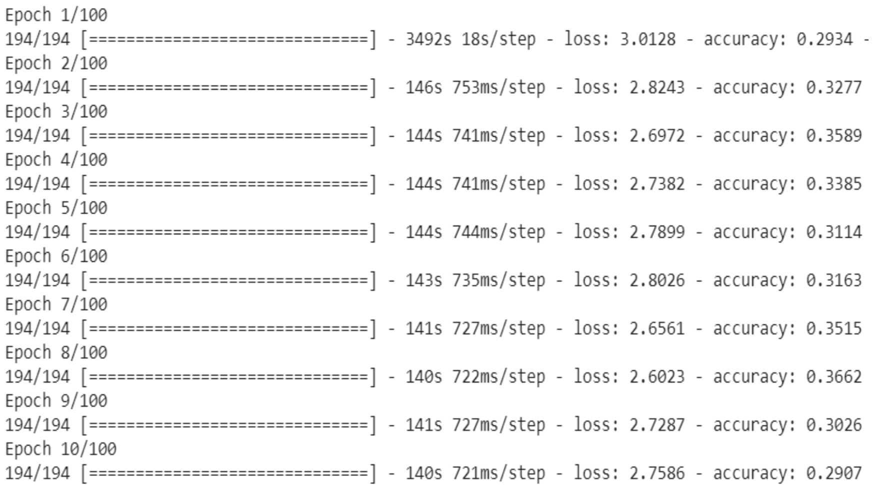
User ID	Software Name	Activation Date	Usage Duration
1001	Photoshop	2023-01-05	2
1002	AutoCAD	2023-02-10	1.5
1003	MS Office	2023-03-15	0.5
1004	Illusrator	2023-04-20	2.5
1005	Quickbooks	2023-05-26	1
1006	MATLAB	2023-06-03	0.8
1007	Solidworks	2023-07-12	2.2
1008	Sketchup	2023-08-08	1.7
1009	Premier Pro	2023-09-20	0.4
1010	Final Cut Pro	2023-10-15	2.3
1011	Norton Antivirus	2023-11-19	1.9
1012	McAfee Security	2023-12-22	0.6
1013	SPSS	2024-01-11	2.4
1014	Oracle DB	2024-02-05	1.6
1015	3ds Max	2024-03-08	0.7

In the context of user behavior, the generator attempts to generate synthetic data based on user behavior patterns [Table 4] that closely resembles original user behavior. The discriminator's task distinguishes between real and generated user behavior. During training, the generator enhances its ability to generate user behavior that is tricky to differentiate from real user data, while the discriminator struggles to become better at telling the difference.

Table 5: Piracy Detection Output

Network Traffic	File Access	Login Attempts	Device Type	Piracy Indicator
150	5	3	Desktop	0
120	10	1	Laptop	0
50	3	2	Desktop	1
180	7	0	Laptop	1
90	8	4	Desktop	1
60	4	1	Laptop	0
170	6	2	Desktop	0
110	9	0	Laptop	0
40	2	3	Desktop	1
160	7	1	Laptop	1
130	11	0	Laptop	0
55	4	2	Desktop	0
175	8	2	Desktop	1

Piracy detection output [Table 5] presents the outcomes generated by the proposed GAN-IF based piracy detection model which is tasked with identifying instances of piracy or any copyright infringement. Piracy detection system is a binary classifier, which categorizes the instances into one of two classes: "Piracy" or "Non-Piracy" which corresponds to its Legitimate or Illegitimate status. While the proposed model makes predictions, it assigns each input either a "Piracy" label indicating potential piracy or a "non-piracy" label indicating that the behavior is genuine and doesn't violate any copyright. Epoch iterations [Figure 5] indicate the number of times the entire dataset has been passed forward and backward through GAN for training. Each iteration results in updating of the model's weights based on the loss estimated from discriminator's capacity to tell apart genuine from created synthetic data.



```
Epoch 1/100
194/194 [==============================] - 3492s 18s/step - loss: 3.0128 - accuracy: 0.2934 -
Epoch 2/100
194/194 [==============================] - 146s 753ms/step - loss: 2.8243 - accuracy: 0.3277
Epoch 3/100
194/194 [==============================] - 144s 741ms/step - loss: 2.6972 - accuracy: 0.3589
Epoch 4/100
194/194 [==============================] - 144s 741ms/step - loss: 2.7382 - accuracy: 0.3385
Epoch 5/100
194/194 [==============================] - 144s 744ms/step - loss: 2.7899 - accuracy: 0.3114
Epoch 6/100
194/194 [==============================] - 143s 735ms/step - loss: 2.8026 - accuracy: 0.3163
Epoch 7/100
194/194 [==============================] - 141s 727ms/step - loss: 2.6561 - accuracy: 0.3515
Epoch 8/100
194/194 [==============================] - 140s 722ms/step - loss: 2.6023 - accuracy: 0.3662
Epoch 9/100
194/194 [==============================] - 141s 727ms/step - loss: 2.7287 - accuracy: 0.3026
Epoch 10/100
194/194 [==============================] - 140s 721ms/step - loss: 2.7586 - accuracy: 0.2907
```

Figure 5: Epochs and Training Loss of GAN Along with Performance Metrics Scores

Table 6: Detected Anomalies in Terms of Pirated Software

User ID	Piracy Label
User ID: 1	0
User ID: 2	0
User ID: 3	1
User ID: 4	0
User ID: 5	1
User ID: 6	0
User ID: 7	0
User ID: 8	0
User ID: 9	1
User ID: 10	0

Piracy Label is either "0" (legal) or "1" (probable piracy) based on the model's predictions. Users with a "Piracy Label" of "1" are those for whom the model has identified potential piracy based on the input features and its learned patterns [Table 6]. These results suggest that our proposed GAN and Isolation Forest-based software piracy detection system has achieved a relatively high level of precision and recall, with a good balance between the two whose confusion matrix is presented in Figure 6.

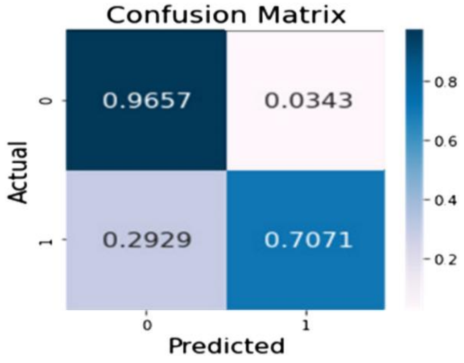


Figure 6: Confusion Matrix

Comparative analysis of our proposed model has been done with One-Class SVM, VAE (Variational Autoencoder), Random Forest, and RNN (Recurrent Neural Network) for software piracy anomaly detection. Below is a tabular comparison of these models across different evaluation metrics (Table 7):

Table 7: Performance Analysis of Proposed GAN-IF with other Techniques

Metrics	One class SVM	VAE	Random Forest	RNN	Proposed GAN-IF based IDS
Accuracy	89.34%	91.33%	88.02%	93.49%	98.50%
Recall	85.92%	92.73%	82.31%	90.27%	92%
Precision	91.36%	89.83%	89.76%	86.92%	95%
F1-score	88.45%	90.24%	86.78%	82.88%	88%
FPR	12.88%	9.67%	14.66%	6.78%	5%
Model complexity	O(n)	O(n2)	O(n2)	O(n3)	O(n)

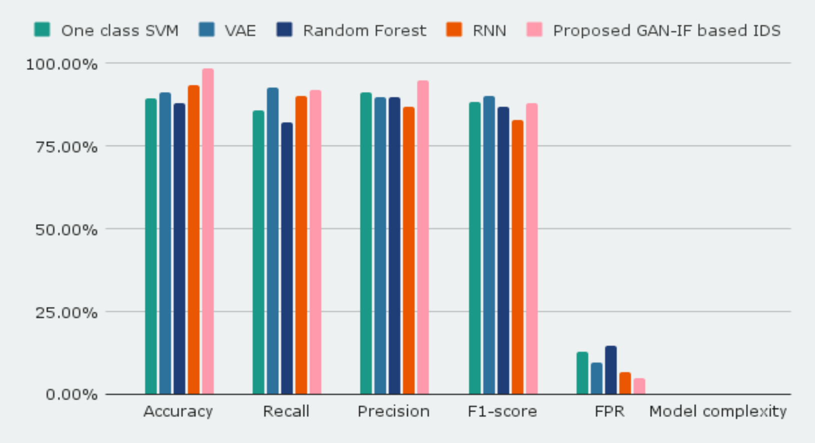


Figure 7: Performance Comparison of Proposed Model

Key Points that have been observed from this comparative analysis [Table 7] are

- Proposed model has achieved the highest accuracy, with RNN slightly outperforming others B [Figure 7].
- VAE and the RNN have the highest recall, indicating their effectiveness in capturing true anomalies. Proposed model also performs well in this regard.
- RNN has the highest F1-score, suggesting a strong balance between precision and recall. Proposed model and VAE also exhibit competitive F1-scores.
- Proposed model has the lowest false positive rate, indicating its ability to minimize false alarms.
- One-Class SVM and Random Forest are relatively simple models to implement, while the RNN is more complex due to its deep learning nature. Proposed model and VAE fall in the moderate complexity range.

With priorities like high accuracy, minimal complexity, faster operating speed and minimization of FPs, GAN-IF model is observed to perform well while having reasonable model complexity and fast operating speed, it can be concluded that the proposed model is a strong candidate for software piracy detection. While GANs have the ability to learn complex patterns in data, making them potentially effective for detecting sophisticated piracy activities, Isolation Forests provide interpretability, as they are efficient and scalable to large datasets, making them highly suitable for real-time applications. This integrated model is a novel solution that combines the power of Generative Adversarial Networks (GANs) with the anomaly detection capabilities of Isolation Forests to effectively identify abnormal patterns in data, which is essential for detecting software piracy attempts or anomalies in software usage.

5 Conclusion

In today's interconnected world, cyber-security is an unending practice that requires continuous observation and adaptation to ever-evolving threat landscapes. It's not just a worry for larger conglomerates but also for individuals who store private and financial data online. By employing robust cyber-security measures, corporate and individuals can minimize their risk of becoming victim to cyber-attacks and safe guard their digital assets. By effective integration of Generative Adversarial Networks (GANs) and Isolation Forests in an Intrusion Detection System (IDS), several advantages in terms of effectiveness, efficiency, versatility and adaptability can be accomplished. GANs excel at generating synthetic data that resembles original network traffic, which enhances the IDS's ability to detect unusual or novel attacks. Isolation Forests, on the other hand, are talented at identifying anomalies within high-dimensional data, making them a suitable choice for network intrusion detection thus handling of evolving attack strategies and reducing false positives is made possible. Through better understanding of detected anomalies and by improving incident responses, this cost-effective solution for cyber-security ensures security and reliability of GAN-generated data in real-world applications. With persistent development of threat landscape in cyber-security arena, our GAN-IF based model symbolizes a momentous stride onward in improving intrusion detection abilities for software piracy, thereby recommending a practical and adaptive defense mechanism for safeguarding IP and digital assets. Furthermore, our future enhancement is focused on developing lightweight and efficient security solutions tailored to meet the demands of resource-constrained IoT devices which is crucial for safeguarding this growing ecosystem along with prioritizing the creation of quantum-resistant cryptographic methods, these threats should also be addressed in the prospective future.

List of Acronyms

AI	-	Artificial Intelligence
CNN	-	Convolutional Neural Networks
DL	-	Deep Learning
DT	-	Decision Trees
GAN	-	Generative Adversarial Networks
IF	-	Isolation forest
IoT	-	Internet of Things
LSTM	-	Long Short-Term Memory
ML	-	Machine Learning
RF	-	Random Forest
RNN	-	Recurrent Neural Network
SVM	-	Support Vector machine
VAE	-	Variable Auto Encoder

References

[1] AbuAlghanam, O., Alazzam, H., Alhenawi, E. A., Qatawneh, M., & Adwan, O. (2023). Fusion-based anomaly detection system using modified isolation forest for internet of things. *Journal of Ambient Intelligence and Humanized Computing*, *14*(1), 131-145.

[2] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), 1-29.

[3] Aldhyani, T.H., & Alkahtani, H. (2023). Cyber Security for Detecting Distributed Denial of Service Attacks in Agriculture 4.0: Deep Learning Model. *Mathematics*, *11*(1), 1-19.

[4] Aldriwish, K. (2021). A deep learning approach for malware and software piracy threat detection. *Engineering, Technology & Applied Science Research*, *11*(6), 7757-7762.

[5] Al-Garadi, M.A., Mohamed, A., Al-Ali, A.K., Du, X., Ali, I., & Guizani, M. (2020). A survey of machine and deep learning methods for internet of things (IoT) security. *IEEE Communications Surveys & Tutorials*, *22*(3), 1646-1685.

[6] Benaddi, H., Jouhari, M., Ibrahimi, K., Benslimane, A., & Amhoud, E.M. (2022). Adversarial Attacks Against IoT Networks using Conditional GAN based Learning. *In GLOBECOM IEEE Global Communications Conference*, 2788-2793.

[7] Chen, D., Wawrzynski, P., & Lv, Z. (2021). Cyber security in smart cities: a review of deep learning-based applications and case studies. *Sustainable Cities and Society*, *66*.

[8] Dina, A.S., & Manivannan, D. (2021). Intrusion detection based on machine learning techniques in computer networks. *Internet of Things*, *16*.

[9] Diro, A.A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, *82*, 761-768.

[10] Grover, M., Sharma, N., Bhushan, B., Kaushik, I., & Khamparia, A. (2020). Malware threat analysis of IoT devices using deep learning neural network methodologies. *Security and Trust Issues in Internet of Things*, 123-143.

[11] Gümüşbaş, D., Yıldırım, T., Genovese, A., & Scotti, F. (2020). A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. *IEEE Systems Journal*, *15*(2), 1717-1731.

[12] Jana, S., Thangam, S., Kishore, A., Sai Kumar, V., & Vandana, S. (2022). Transfer learning based deep convolutional neural network model for pavement crack detection from images. *International Journal of Nonlinear Analysis and Applications*, *13*(1), 1209-1223.

[13] Khan, M.A., Khan, M.A., Jan, S.U., Ahmad, J., Jamal, S.S., Shah, A.A., & Buchanan, W.J. (2021). A deep learning-based intrusion detection system for MQTT enabled IoT. *Sensors*, *21*(21), 1-25.

[14] Kim, J., Kim, J., Kim, H., Shim, M., & Choi, E. (2020). CNN-based network intrusion detection against denial-of-service attacks. *Electronics*, *9*(6), 1-21.

[15] Kumaran, U. (2021). A secure and privacy-preserving approach to protect user data across cloud based online social networks. In *Research Anthology on Artificial Intelligence Applications in Security*, 560-585. IGI Global.

[16] Nagarajan, S.M., Anandhan, P., Muthukumaran, V., Uma, K., & Kumaran, U. (2022). Security framework for IoT and deep belief network-based healthcare system using blockchain technology. *International Journal of Electronic Business*, *17*(3), 226-243.

[17] Otoum, S., Kantarci, B., & Mouftah, H.T. (2019). On the feasibility of deep learning in sensor network intrusion detection. *IEEE Networking Letters*, *1*(2), 68-71.

[18] Otoum, Y., Liu, D., & Nayak, A. (2022). DL-IDS: a deep learning–based intrusion detection framework for securing IoT. *Transactions on Emerging Telecommunications Technologies*, *33*(3), 1-16.

[19] Pinto, L., Brito, C., Marinho, V., & Pinto, P. (2022). Assessing the Relevance of Cybersecurity Training and Policies to Prevent and Mitigate the Impact of Phishing Attacks. *Journal of Internet Services and Information Security (JISIS)*, *12*(4), 23-38.

[20] Piplai, A., Chukkapalli, S.S.L., & Joshi, A. (2020). NAttack! Adversarial Attacks to bypass a GAN based classifier trained to detect Network intrusion. *In IEEE 6th Intl Conference on Big Data Security on Cloud (Big Data Security), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, 49-54.

[21] Sadaf, K., & Sultana, J. (2020). Intrusion detection based on autoencoder and isolation forest in fog computing. *IEEE Access*, *8*, 167059-167068.

[22] Saheed, Y.K., Abiodun, A.I., Misra, S., Holone, M.K., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. *Alexandria Engineering Journal*, *61*(12), 9395-9409.

[23] Seo, E., Song, H.M., & Kim, H.K. (2018). GIDS: GAN based intrusion detection system for in-vehicle network. *In IEEE 16th Annual Conference on Privacy, Security and Trust (PST)*, 1-6.

[24] Shankar, S., & Thangam, S. (2019). Integrated system for easier and effective access to drug information. *Biomedical and Pharmacology Journal*, *12*(3), 1069-1077.

[25] Sharma, B., Sharma, L., Lal, C., & Roy, S. (2023). Anomaly based network intrusion detection for IoT attacks using deep learning technique. *Computers and Electrical Engineering*, *107*.

[26] Sultana, N., Chilamkurti, N., Peng, W., & Alhadad, R. (2019). Survey on SDN based network intrusion detection system using machine learning approaches. *Peer-to-Peer Networking and Applications*, *12*, 493-501.

[27] Tao, X., Peng, Y., Zhao, F., Zhao, P., & Wang, Y. (2018). A parallel algorithm for network traffic anomaly detection based on Isolation Forest. *International Journal of Distributed Sensor Networks*, *14*(11), 1-11.

[28] Tian, Z., Luo, C., Qiu, J., Du, X., & Guizani, M. (2019). A distributed deep learning system for web attack detection on edge devices. *IEEE Transactions on Industrial Informatics*, *16*(3), 1963-1971.

[29] Van, N.T., & Thinh, T.N. (2017). An anomaly-based network intrusion detection system using deep learning. *In IEEE international conference on system science and engineering (ICSSE)*, 210-214.

[30] Verma, A., & Ranga, V. (2020). Machine learning based intrusion detection systems for IoT applications. *Wireless Personal Communications*, *111*, 2287-2310.

Authors Biography

Kumaran U (Member, IEEE) received the master’s and Bachelor’s degrees in computer science and engineering from Arunai Engineering College, affiliated to Anna University, Chennai, India, and the Ph.D. degree in computer science from the Vellore Institute of Technology, Vellore, India in June 2020. He is currently an Assistant Professor (SG) with Computer Science and Engineering Department, Amrita School of Computing, Bengaluru, India. He has more than 14 years of teaching experience and 10 years of research experience in the domain of computer science and engineering. He is currently guiding five PhD scholars and two M.tech students. His main research interests include Machine Learning, Cybersecurity, Internet of Things, Cloud computing and Privacy Preserving issues and data mining.

Dr. Thangam S, BE, ME, Ph.D., Dr. Thangam S. currently serves as an Assistant Professor (SG) at the Amrita School of Computing, Bengaluru.,Amrita Vishwa Vidyapeetham, India. Her area of interest in research includes Service Oriented Architecture, Networks, IoT, Artificial Intelligence, Data Structures and Cloud Computing. She completed her Ph. D. in Computer Science and Engineering from Anna University, Chennai. India. She has 23 years of teaching experience. She has published her research works in 25 international journals and conferences. She is a member of ISTE.

Dr. Nidhin Prabhakar T V, B.Tech, M.Tech, Ph.D., Dr. Nidhin Prabhakar T V currently serves as an Assistant Professor at the Amrita School of Computing, Bengaluru.,Amrita Vishwa Vidyapeetham, India. His research interest includes Image processing, Machine learning, Deep learning, Remote sensing, Computer vision and Artificial Reality/Virtual Reality (AR/VR). He completed his Ph. D in Engineering from Amrita Vishwa Vidyapeetham, Coimbatore. India. He has more than 2 years of teaching experience. He has published his research works in 12 international journals and conferences.

Jana Selvaganesan received the BE degree in electronics and communication engineering from Bharathidasan University, the ME degree in communication systems from Anna University Chennai, and the PhD degree in information and communication engineering from Anna University. She has over 25 years of academic experience and is currently working as a professor in the Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology. She is a member of ISTE and IEEE Signal Processing Society. She has published over 25 papers in national and international conferences and journals. Her research interests include artificial intelligence, machine learning, deep learning, signal processing, and computer vision.

Mr. Vishwas H.N., B.E, M.Tech. currently serves as an Assistant Professor (Sr. Gr.) at Amrita School of Computing, Bengaluru., Amrita Vishwa Vidyapeetham, India. His research interest includes Internet of Things, Cybersecurity, WSN, Switching and Routing, and Machine learning. He has more than 10 years of teaching experience and 5 years of research experience in the domain of computer science and engineering. He has an additional designation as Instructor, at Cisco Networking Academy at Amrita Bangalore Campus. He also has published his research works in 12 reputed international conferences/journals.