**Detection of Real-Time Malicious Intrusions and Attacks in IoT Empowered Cybersecurity Infrastructures**

**Abstract:**

In the realm of cybersecurity, the proliferation of computer viruses, malicious activities, and hostile attacks poses a significant threat to computer networks. Intrusion detection stands as a crucial element in network security, serving as an active defense technology. Traditional intrusion detection systems encounter challenges such as poor accuracy, ineffective detection, a high rate of false positives, and a limited capability to adapt to emerging intrusion types. To overcome these issues, we propose a novel method based on deep learning for detecting cybersecurity vulnerabilities and breaches in cyber-physical systems. Our proposed framework distinguishes itself by employing unsupervised and deep learning-based discriminative approaches. Specifically, we introduce a Generative Adversarial Network (GAN) tailored for detecting cyber threats in Internet of Things (IoT)-driven Industrial Internet of Things (IIoT) networks. Additionally, our framework ensures the confidentiality and integrity of users' and systems' sensitive information throughout the training and testing phases. The utilization of Bidirectional Long Short-Term Memory (BiLSTM) networks contributes to the understanding of sequential data for classify the attack . We validate the effectiveness of our approach using diverse datasets, including NSL-KDD. Furthermore, our framework demonstrates superior outcomes when compared to well-established state-of-the-art deep learning classifiers.

**Existing system:**

Existing work has integrated Generative Adversarial Networks (GANs) with Recurrent Neural Networks (RNNs) to enhance the capabilities of cyber threat detection in various domains. This integration harnesses the strengths of GANs in generating realistic data samples and RNNs in capturing sequential dependencies. By combining these two architectures, the system can effectively model and detect complex patterns in time-series data, such as network traffic in IoT-driven Industrial Internet of Things (IIoT) networks.

**Drawback:**

1. **Increased Training Complexity**: Combining GANs with RNNs raises the complexity of model training due to the need for tuning multiple hyperparameters and managing the training dynamics of both architectures.
2. **Mode Collapse and Stability Concerns**: There's a risk of mode collapse with GANs, which can be amplified when integrated with RNNs, potentially leading to unstable training and reduced effectiveness in detecting cyber threats.

**Proposed System:**

The proposed system is a cybersecurity framework designed specifically for Internet of Things (IoT)-driven Industrial Internet of Things (IIoT) networks. It employs a combination of unsupervised and deep learning-based discriminative approaches, including a specialized Generative Adversarial Network (GAN), to detect cyber threats effectively. The GAN is tailored for cybersecurity purposes within IIoT networks. GANs consist of two neural networks, a generator and a discriminator, which compete against each other to generate realistic data samples and detect anomalies respectively. In this context, the GAN is trained to generate fake data for related to attack. BiLSTM networks are employed to understand sequential data, which is common in network traffic analysis. BiLSTM networks are capable of capturing dependencies and patterns in sequential data, making them well-suited for classifying cyber attacks based on their temporal characteristics. The system classifies detected anomalies and threats based on their characteristics and behaviors. This classification enables rapid response and mitigation strategies to be deployed, minimizing the potential impact of cyber attacks on IIoT networks.

**Advantage:**

1. **High Accuracy**: The system offers superior threat detection accuracy in IoT-driven IIoT networks, thanks to its utilization of advanced deep learning techniques like GANs and BiLSTM networks.
2. **Continuous Adaptation**: It continuously learns from new data and adapts to evolving cyber threats, ensuring sustained effectiveness in detecting emerging attack vectors and maintaining robust security in IIoT environments.

**Hardware Requirements**

* System: Intel core I3 3.80 GHz 64 bit.
* Monitor: LED.
* Mouse: Logitech.
* Ram: 4.00 GB.

**Software Requirements:**

* Operating system : Windows
* Language : Python
* Platform : Anaconda 3

**Modules:**

* Dataset Collection
* Preprocessing
* GAN
* **BiLSTM**
* Novel Attack Detection System

**Modules Description:**

Dataset Collection:

In this module we collect the NSL-KDD dataset , which contain (1) Denial-of-Service (DoS) attacks, (2) unauthorized access from a remote machine (R2L) attacks, unauthorized access to local superuser with root privileges (U2R) attacks, and surveillance and other probing attacks.

**Preprocessing:**

The proposed schema for data preprocessing majorly consists of two phases, primarily OneHotEncoder (OHE) and min-max normalization. The onehotencoder to the integer’s representation. This is where the encoded integer variable is extracted and for each unique integer value, a new binary variable is inserted.

Steps:

1. missing\_values (identify missing values from the dataset )
2. strategy # mean or median or most frequent values are used as the strategic replacement of missing value
3. Encoding categorical data # apply data transformation techniques, onehotencoder is applied to the categorical data
4. Splitting the Dataset into the Training set and Test Set (A general rule of the thumb is to allocate 80% of the dataset to the training set and the remaining 20% to test set)
5. Feature Scaling with Normalization (min-max normalization)

**GAN:**

Only normal data patterns are used for training. The GAN generator and discriminator are trained and then the Encoder is trained within the autoencoder architecture by using the trained generator as the Decoder**.** Generative Adversarial Networks are powerful modeling frameworks for high-dimensional data that build two competing networks: a generator G and a discriminator D. The generator network is trained to produce synthetic data examples that are similar to real data patterns by taking a random vector z, drawn from an input distribution Pz(z) in a latent Z-Space. If trained only with normal data patterns, the generator captures the hidden multivariate distribution of the training sequences and can be seen as an implicit model of the system at normal status. On the other hand, the discriminator network is trained to distinguish between the generated synthetic examples and real data patterns, and then classify data patterns in one of these two classes.

**Autoencoder:**

The discriminator D has its weights initialized with the Xavier approach, and is trained with the Gradient Descent Optimizer to minimize the mean negative cross-entropy between its predictions and sequence labels. The autoencoder configuration and is obtained from the training of an autoencoder. The Encoder part of the autoencoder maps input data into the latent space. The Decoder part, on the other hand, corresponds to the GAN generator, which reconstructs the data from its representation in the latent space by performing the mapping. . Thus, it is trained by minimizing the mean squared error (MSE) residual loss between the input data x and reconstructed data xnew.

**Bidirectional Long Short-Term Memory (BiLSTM) Networks**:

In this module involves several key steps to ensure the effectiveness of the classification module. Initially, the dataset is divided into training, validation, and testing sets to facilitate model training, validation, and evaluation. The splitting ensures that the model learns from a diverse range of data while also assessing its generalization ability on unseen samples. Subsequently, the training phase commences, where the BiLSTM network is trained on the training set using backpropagation through time (BPTT) to minimize the classification error. During training, the network adjusts its parameters iteratively to better capture the sequential dependencies and patterns present in thedataset, thereby improving its ability to classify cyber threats accurately. Furthermore, to prevent overfitting and ensure optimal model performance, the validation set is employed for hyperparameter tuning and early stopping based on performance metrics such as validation loss or accuracy

**Novel Attack Detection System:**

In the testing phase, the trained Bidirectional Long Short-Term Memory (BiLSTM) network is evaluated using a separate dataset not encountered during training or validation. Each data instance in the testing dataset is inputted into the network, which predicts class labels or probability distributions for different cyber threat categories. Performance metrics like accuracy, precision, recall, F1-score are calculated by comparing these predictions against ground truth labels. The testing phase validates the model's ability to accurately classify cyber threats, providing insights into its generalization ability and real-world effectiveness.

Architecture Diagram:

Data Set collection

Pre-processing

GAN Based malicious Data Generation

GAN Network Training

Classifier

Generator

Generate Fake Malicious Data And make Final Dataset

Generate Malicious Traffic data

Dataset Splitting

BILSTM Training

BILSTM Testing

Performance Evaluation

**Flow Diagram:**

Dataset Collection

Preprocessing

GAN Fake Data Generation

Make Final Dataset

Dataset Splitting

BILSTM Training

Testing