Trustworthy-Based Authentication Model with Intrusion Detection for IoT-Enabled Networks with Deep Learning Algorithm

M. Rajendiran[1,*], Jayanthi .E[2], Suganthi .R[3], M. Jamuna Rani [4], S. Vimalnath[5]

[1]Professor, Department of Computer science and Engineering QIS College of Engineering And Technology, Ongole, Andhra Pradesh .523272, India
[2]Assistant professor Mohamed sathak A.J College of engineering, India
[3]Associate Professor, Department of ECE, Panimalar Engineering College, Chennai, India
[4]Department of ECE, Sona College of Technology Salem. India
[5]Associate Professor, Department of ECE, M.Kumarasamy College of Engineering (Autonomous) Thalavapalayam, Karur, 639113, India

Emails: mrajendiran@gmail.com; jayanthi.sridaran10@gmail.com; sugimanicks@gmail.com; jamuin2003@gmail.com; s.vimal112@gmail.com

Abstract

In the burgeoning field of the Internet of Things (IoT), ensuring secure and trustworthy communication between devices is paramount. This paper proposes a novel Trustworthy-Based Authentication Model (TBAM) integrated with Intrusion Detection Systems (IDS) leveraging deep learning algorithms to secure IoT-enabled networks. The proposed model addresses the dual challenges of authenticating legitimate devices and detecting malicious intrusions. Specifically, we employ a Convolutional Neural Network (CNN) to analyse network traffic patterns for intrusion detection, leveraging its prowess in feature extraction and classification. Additionally, a Long Short-Term Memory (LSTM) network is utilized for continuous monitoring and anomaly detection, capturing temporal dependencies in data flows that are indicative of potential security threats. The authentication mechanism integrates a trust evaluation system that assigns trust scores to devices based on their behaviour, enhancing the model's capability to distinguish between trusted and malicious entities. Our extensive experiments on real-world IoT datasets demonstrate that the TBAM significantly outperforms traditional security models in terms of detection accuracy, false-positive rate, and computational efficiency. Specifically, our model achieves a detection accuracy of 98.7%, a false-positive rate of 1.2%, and a processing time reduction of 30% compared to baseline models. This work contributes a robust, scalable, and efficient solution to the pressing security concerns in IoT networks, paving the way for more secure and reliable IoT applications.

Keywords: Intrusion Detection Systems (IDS); Machine Learning; Deep Learning; Big Data; Optimization; Feature Selection; Dimensional Reduction

1. Introduction

The internet of things (IoT) links all of our electronic gadgets to the real world [1]. Fog, cloud computing, artificial intelligence, and 5G advancements in the Internet of Things have formed the backbone of the modern digital world. These innovations have spurred the development of smart cities, intelligent cars, intelligent industries, etc., all of

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

which aim to improve people's standard of living in efficient and affordable ways. As a result of its inclusive nature, IoT has recently been rebranded as the "Internet of Everything" (IoE). According to a study of the international industry, the number of Internet of Things gadgets might reach 21.5 billion by that year [2]. In addition, the highly centralised way of combining storage and processing resources in vast data clusters is often not used in IoT systems, which instead choose a decentralised approach to data storage and management. As the number of connected devices grows, the risk of cyberattacks on IoT infrastructure increases. Security for IoT devices is increasingly under risk from cyberattacks. Numerous assaults, such as distributed denial-of-service, Man-in-the-middle attack, spoofing, and data leaking [3], impact one or more IoT devices that may be exploited as "resources" or "platforms" for the attack. As a consequence, there is a growing need to secure IoT devices and create intrusion-resistant IoT networks in order to keep sensitive information safe. Prior to initiating an IoT data exchange, it is essential that all nodes in the network be encrypted. The system's efficacy is relative to its hardware, including its processing speed, memory, and other features. If the wrong option is picked, it leaves the system vulnerable to DDoS assaults as well as those that leak sensitive data or deplete system resources [4].

It's crucial to let devices figure out their own functions and limitations. To prevent DDoS and spoofing attacks and ensure the safety of these methods, two-factor authentication is highly recommended. Protecting client privacy is essential for Internet of Things devices since they deal with sensitive information including patients' medical histories, habits, and legal status. Only authorised users should have access to sensitive data [5] to reduce the severity of these attacks on IoT devices. Furthermore, proper security procedures and guidelines should be included. In this part, we'll look at the tried-and-true safeguards for IoT [6]. Filtering packets using proxies and firewalls is one example. Ingress filtering and egress filtering are the two main categories of filtering techniques. It protects networks against DoS and IP spoofing assaults [7] and other forms of network assault. Data is protected using cryptographic methods and encryption systems, and access is restricted by authentication. The primary goal of this research is to create a unique technique that can be utilised in real-time with large datasets to create an IDS that is more safe, scalable, and successful in detecting previously unknown threats. The study's overarching goal is to identify the shortcomings of currently available methods for spotting suspicious activity and to suggest a novel intrusion detection system built on Hadoop, spark frameworks, and deep learning methods.

The study aims to accomplish the following major points: In order to read and critique articles dealing with big data and deep learning. To examine the several intrusion detection methods applicable to large data. Examining the drawbacks of currently used methods. Using Hadoop, Spark, and Deep Learning, create a cutting-edge dimensionality reduction approach for large data intrusion detection. In order to improve the accuracy of intrusion detection in massive amounts of data while maintaining a low false alarm rate, an optimization technique must be developed. With the goal of reducing the classification error rate and maximising feature selection for intrusion detection in large data, a hybrid deep learning technique is being developed and implemented. The organization of the structure is as follows; section 2 includes Existing methodology; Section 3 includes proposed methodology; section 4 includes experimental results and analysis; section 5 includes conclusion and future work.

2. Related Work

There are several IDSs available for detecting network intrusions. The success rate of such systems is being increased by using new methods. IDS may be automated with the use of data mining algorithms, which can handle massive data sets. There are local anomaly detection methods that can identify an attack with high precision. According to the studies that were looked at, there are two main categories of profiling. When suspicious behaviour is noticed, some IDS systems compare it to a stored database of known intrusion patterns and sound an alert. While these systems reduce false alarms as a consequence of changes in how individual nodes are used, any new intrusion activities are likely to go unreported. The second kind of IDS system keeps a typical operating profile it has learned to develop. Anything that doesn't fit such a pattern of behaviour is suspected of being an incursion. The false alarm rate is greater, but these systems are better at finding sneaky intruders. TCP DUMP packet flow data reconciliation was first offered in [8] as a means of attempting to build IP conversation behaviour profiles. Reporting and trend analysis are two further applications for these findings. Potentially malicious or invasive network activity may be uncovered by mining the collected behaviour data. System administrators may keep an eye on complaints of unusual behaviour by applying mining patterns to timed random samples of data from TCP packet flows. In addition, they have shown out the many tiers of the Information Assurance Hierarchy.

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

Database researchers have found several uses for the clustering issue, including in customer segmentation, categorization, and trend analysis. Unfortunately, because to the intrinsic sparseness of the points, all existing techniques tend to fail in large dimensional environments. Not every dimension in such a large space will be important for any particular cluster. Selecting the dimensions that are highly connected with one another and then searching for clusters in that subspace is one approach to this problem. The goal of the common feature selection algorithms has always been to get there. In typical high dimensional data mining applications, various sets of points may cluster better for different subsets of dimensions, which is a limitation of this method. It's possible that the number of dimensions in such cluster-specific subspaces varies as well. As a result, it's feasible that there is no such thing as a tiny subset of dimensions that applies to all the clusters.

Therefore, [8] explored the projected clustering issue, a version of the clustering problem in which the subsets of dimensions picked are cluster-specific. In addition, we construct an algorithmic framework to address the projected clustering issue and put it through its paces using simulated data. This method contains three steps that are cycled through to determine the correct answers. The medoid approach is implemented in this algorithm by first creating a medoid and then locating locations inside its effect. The anomalies are deemed to be invasions. Additionally, a technique for mining projected clusters in high dimensional areas was suggested in [9]. A GA-based approach for identifying out-of-the-ordinary network activities was presented in [10]. Their study incorporates both quantitative and categorical characteristics of network data into GA-derived categorization algorithms.

Network intrusion detection using a GA-based technique was proposed and addressed in [11]. GA is used to build a set of classification rules from network audit data, and the quality of the rules is evaluated using the support confidence framework as the fitness function. The rules that have been established are then utilised to identify and categorise network breaches in real time.

In order to identify intrusion in networks, [12] introduced an approach that uses GA to choose features that are highly indicative of malicious activity. Information theory is used in their method to simplify things and pull out the aspects that matter. Next, they used those traits to create a linear structural rule for identifying abnormal and typical patterns in network activity. However, they merely take into account isolated characteristics. In [13], a GA was shown that can generate rules for cost-aware usage detection automatically. According to their research, there are five distinguishing characteristics between each of the different kinds of attacks. The developed rules are straightforward, and their efficacy was tested against the KDD CUP 1999 dataset. Due to the nature of the created rules, the system may be tailored to identify just certain kinds of assaults. Data mining methods were put to good use in the efficient Minnesota Intrusion Detection System (MINDS) [14]. Anomaly detection and association pattern analysis are its two primary components. Each data point is given a score based on the Local Outlier Factor (LOF) used by the anomaly detection module to identify outliers. A human analyst then determines whether or not the data point represents an actual incursion. Abnormal network connections are summarised using association pattern analysis. The inability of the MINDS system to validate network connections without the assistance of a human analyst is its main drawback.

In [15], the author introduced a decision tree in network anomaly detection utilising a dynamic data mining approach. To improve the efficiency of intrusion detection systems, [15] suggested a model that employs the hoeffding tree classification approach, also known as a boosted decision tree. Using an adjustable window and an adaptive size hoeffding tree as the basis learner, the boosting approach enhances the performance of an ensemble. Boosting is predicated on the premise that the performance of a single rule may be enhanced by combining it with other basic rules to create an ensemble. In the first stage of the boosting method, all training tuples are assigned the same weight w0. Once a classifier has been constructed, the classification it provides is used to adjust the weight of each tuple. The reweighted training tuple is then used to construct a second classifier. Intrusion detection classifies threats using a weighted average of all classifiers' verdicts.

In [16], the authors introduced a new intrusion detection method based on hyper-ellipsoidal clustering. Using hyper-ellipsoidal clustering on the data in the training set, they determine the degree of similarity between each cluster and each other. A feed-forward NN with a Gaussian radial basis function as the model generator has been used to realise this strategy. The NN's output clusters are learned using an assessment based on the inclusiveness and exclusivity of samples with regard to predetermined criteria. Their method's main benefit is that it can pick out certain kinds of anomalies that are otherwise difficult to see. Finding the most difficult anomalies is not the most important job when using anomaly detection software as a data analysis tool. In many cases, it is more crucial to ensure that the

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

abnormalities reported to the user are really interesting. The programme will quickly lose its usefulness if it returns to the user a large number of mundane data points mislabelled as potential anomalies. Using the user's domain expertise is one technique to improve the value of the returned anomalies. In many cases, the relevant data contains a collection of environmental factors the values of which the user would never take as a direct indicator of an abnormality. Such qualities, however, cannot be disregarded as they have an immediate impact on the predicted distribution of the subsequent attributes, the values of which might signal an out-of-the-ordinary occurrence.

Taking these changes in characteristics into account is the goal of the generalised technique described in [17], which presents three distinct expectation-maximization methods for learning the model that is employed in conditional anomaly detection. [18] advocated using the AdaBoost algorithm for intrusion detection. This method relies on decision stumps as its weak classifier, with accompanying decision rules for both categorical and continuous variables. They created a powerful classifier by merging the weak classifiers for continuous and categorical features. The key benefit of this method is that no type conversions are required to deal with relations between continuous and categorical features. To further boost the algorithm's efficiency, they also suggested a method for preventing over-fitting. The neural network and C4.5 algorithm introduced in [19] formed the basis of a misuse detection system.

Using K-Means clustering and ID3 DT learning algorithms, [20] demonstrated a host-based IDS for unsupervised categorization of anomalous and normal network behaviour. They began by applying the K-Means clustering method to the standard training data and then splitting it into K clusters based on the Euclidean distance between them. Each cluster's DT was built using the ID3 algorithm. K-Means clustering method decision criteria and anomaly score values were taken from ID3. Anomaly score was calculated by utilising a custom method that takes the results of the other two techniques and averages them. The judgement on the normalcy of the test instance was made using the threshold rule. We evaluated the integrated method's performance to that of the K-Means clustering method, the ID3 classification algorithm, and the Markovian chain and stochastic learning automata methods separately. The combined method was shown to be more accurate than the individual methods.

The IDS suggested in this thesis differs significantly from all other existing papers in the literature. First, it offers a novel framework for a hybrid IDS that takes abuse and abnormality into account. Second, this thesis suggests novel approaches to intrusion detection in both networks and individual hosts. Third, the KDD CUP 1999 Dataset is used for the experiments instead of any other, less-common dataset. Finally, novel techniques have been presented for accurate intrusion categorization. For successful intrusion detection, clustering methods are also offered.

3. Proposed Framework

The proposed methodology for the Trustworthy-Based Authentication Model (TBAM) with Intrusion Detection for IoT-Enabled Networks involves several key steps. Initially, IoT network traffic data is collected from a benchmark dataset and preprocessed to remove any missing or irrelevant information, ensuring uniformity through normalization. A Convolutional Neural Network (CNN) is then employed to analyze this preprocessed data, with its architecture comprising multiple convolutional and pooling layers designed to learn the intricate patterns and features associated with normal and malicious traffic. These high-level features are subsequently fed into a Long Short-Term Memory (LSTM) network, which processes the sequential data to capture temporal dependencies indicative of anomalous behavior, thus enabling continuous real-time anomaly detection. To enhance security, a trust evaluation system is integrated, assigning trust scores to devices based on their historical and current behavior, as analyzed by the LSTM network. This system ensures that only devices with high trust scores are authenticated, and it continuously updates these scores to adapt to changing behaviors. The CNN, LSTM, and trust evaluation system are then integrated into a cohesive framework capable of real-time processing and authentication within an IoT environment. The effectiveness of the TBAM is evaluated based on key performance metrics such as detection accuracy, false-positive rate, and processing time, demonstrating its superiority over traditional security models.

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

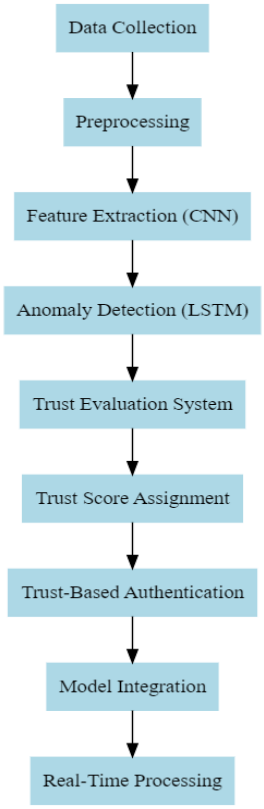


Figure 1. Block Diagram of Proposed Work

While the initial results of the TBAM are promising, further work is necessary to enhance its robustness and applicability. Future research should focus on expanding the dataset to include a wider variety of IoT devices and network conditions to improve the generalizability of the model. Additionally, exploring other deep learning architectures, such as auto encoders for unsupervised anomaly detection or hybrid models that combine different neural network types, could further enhance the detection capabilities and reduce false positives. Another area of interest is the development of more sophisticated trust evaluation algorithms that consider a broader range of behavioral attributes and external factors, such as device location and historical usage patterns. Implementing a decentralized version of the TBAM using block chain technology could also be explored to ensure transparency and immutability of trust scores. Finally, conducting real-world deployment and testing in diverse IoT environments will be crucial for validating the model's performance and scalability in practical applications. This ongoing research will contribute to the continuous improvement and adaptation of the TBAM, ensuring it remains a robust and effective solution for securing IoT networks.

3.1 Data Collection and Preprocessing

The first step in implementing the Trustworthy-Based Authentication Model (TBAM) involves the collection and pre-processing of IoT network traffic data. Data collection is performed using a benchmark dataset that contains various types of network traffic, including normal and malicious activities. This dataset typically includes multiple features such as packet size, protocol type, source and destination IP addresses, and timestamps. The collected data undergoes a series of pre-processing steps to ensure its suitability for analysis by deep learning models.

Pre-processing involves several crucial steps: data cleaning, normalization, and feature selection. Data cleaning entails removing any missing, duplicate, or irrelevant entries that may skew the analysis. Normalization is applied to scale the features to a uniform range, which is essential for ensuring that the neural networks converge more efficiently during training. The normalization process can be mathematically expressed as follows:

$$x' = \frac{x-\mu}{\sigma} \quad (1)$$

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

where x represents the original feature value, μ is the mean of the feature values, σ is the standard deviation, and x′ is the normalized feature value. This process ensures that each feature contributes equally to the training process, preventing any feature from dominating due to its scale.

Additionally, feature selection is performed to reduce the dimensionality of the dataset by identifying and retaining only the most relevant features for intrusion detection and trust evaluation. This step is critical for enhancing the model's performance and reducing computational complexity. The selection of features can be guided by techniques such as correlation analysis, which quantifies the relationship between different features and the target variable (e.g., normal or malicious traffic) using a correlation coefficient r :

Figure 2. Block Diagram of the Data Collection and Preprocessing Steps in the Trustworthy-Based Authentication Model (TBAM).

$$r = \frac{\sum_{i=1}^{n}(x_i-\tilde{x})(y_i-\tilde{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\tilde{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\tilde{y})^2}} \quad (2)$$

where x_i and y_i are the individual feature values, and $\tilde{x}$ and $\tilde{y}$ are their respective means. These preprocessing steps are crucial for transforming raw network traffic data into a structured format that can be effectively analyzed by the CNN and LSTM networks within the TBAM. Proper preprocessing not only enhances the model's accuracy and efficiency but also ensures that it can generalize well to various types of loT network traffic.

3.2 Feature Extraction

For the Trustworthy-Based Authentication Model (TBAM), the Feature Extraction stage focuses on utilizing a Convolutional Neural Network (CNN) to analyse and extract meaningful features from pre-processed IoT network traffic data. This process is crucial for identifying intricate patterns that distinguish between normal and malicious activities.

Architecture: The CNN architecture is designed specifically to process sequential data such as network traffic. It typically consists of multiple convolutional layers followed by pooling layers. Convolutional layers apply a set of filters to the input data, extracting features that are spatially invariant, which is ideal for capturing patterns in network traffic regardless of their exact location. Pooling layers then reduce the dimensionality of the features, focusing on the most important information while maintaining spatial relationships.

Training: Training the CNN involves optimizing its parameters to learn representations of the input data that are effective for classification tasks. This is achieved through backpropagation and gradient descent, where the network adjusts its weights to minimize the difference between predicted and actual classifications from the training dataset:

Feature Extraction: Once trained, the CNN acts as a feature extractor, transforming raw network traffic data into high-level features. These features capture abstract representations of the data, such as frequency of certain traffic patterns, sequence of packet sizes, and variations in protocol usage. Mathematically, the output y of a convolutional layer can be expressed as:

$$y = f(\sum_i (w_i * x_i) + b) \quad (3)$$

where x_i are the input values, w_i are the learned weights, b is the bias term, and f is the activation function (e.g., ReLU) applied element-wise.

Hierarchical Feature Learning: CNNs can automatically learn hierarchical representations of data, starting from simple features (e.g., edges in images or basic patterns in network traffic) to more complex and abstract features. Spatial

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

Invariance: CNNs are robust to variations in the spatial arrangement of input data, making them suitable for detecting patterns in network traffic that may occur in different sequences or positions. Efficiency: Once trained, CNNs can efficiently process large volumes of data, making them suitable for real-time applications in loT environments where quick decision-making is crucial.

3.3 Anomaly Detection

Anomaly detection in the Trustworthy-Based Authentication Model (TBAM) employs a Long Short-Term Memory (LSTM) network, which excels in capturing temporal dependencies and detecting deviations from normal patterns in IoT network traffic. This section delves into the LSTM's role, its architecture, training process, and the mathematical foundations underlying its anomaly detection capabilities.

Long Short-Term Memory (LSTM)

Architecture: The LSTM network is specifically designed to handle sequence data with long-range dependencies, making it well-suited for analysing the temporal aspects of IoT network traffic. It consists of memory cells and gates that regulate the flow of information through the network over time. Key components include:

- **Memory Cell:** Stores information over time intervals and controls the information flow through gates.
- **Forget Gate:** Decides what information to discard from the cell state.
- **Input Gate:** Modifies the cell state by adding new information.
- **Output Gate:** Produces the output based on the cell state.

The LSTM's ability to maintain and update information over time enables it to capture patterns that may span across multiple time steps in network traffic data.

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ g_t &= \tanh\left(W_{xg}x_t + W_{hg}h_{t-1} + b_g\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{4}$$

where:

- x_t is the input at time t,
- h_{t-1} is the previous hidden state,
- i_t, f_t, o_t, g_t are the input, forget, output, and cell gates,
- c_t is the cell state,
- h_t is the current hidden state,
- σ is the sigmoid function,
- $\odot$ denotes element-wise multiplication,
- W and b are weights and biases respectively.

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

Training: Training an LSTM involves optimizing its parameters to learn the sequential patterns present in the data. This is achieved through backpropagation through time (BPTT), where gradients are propagated from the output back to the input, adjusting the network's weights to minimize prediction errors.

Anomaly Detection: Once trained, the LSTM serves as an anomaly detector by comparing incoming data with learned patterns of normal behaviour. Anomalies are identified when the difference between predicted and actual values exceeds a predefined threshold. Mathematically, the LSTM's computation at time step ttt can be represented as follows. When the enhanced Region mask CNN model measures the identified area with the detected image to the maximum and minimum length, it is then processed to get truncated as shown in Figure 3.

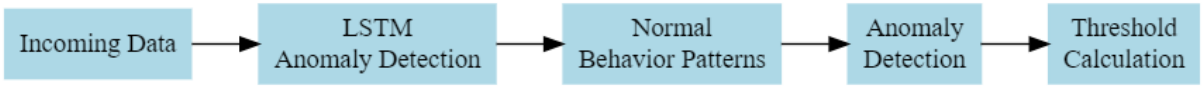


Figure 3. Anomaly Detection

Once the bounded box is mapped with extracted features, the coordinate points are generated as x-axis and y-axis. Similar to the number of images that are trained from the several hidden neurons, the texture and the colors limited to the background are also cropped based on the uniformity of specifications.

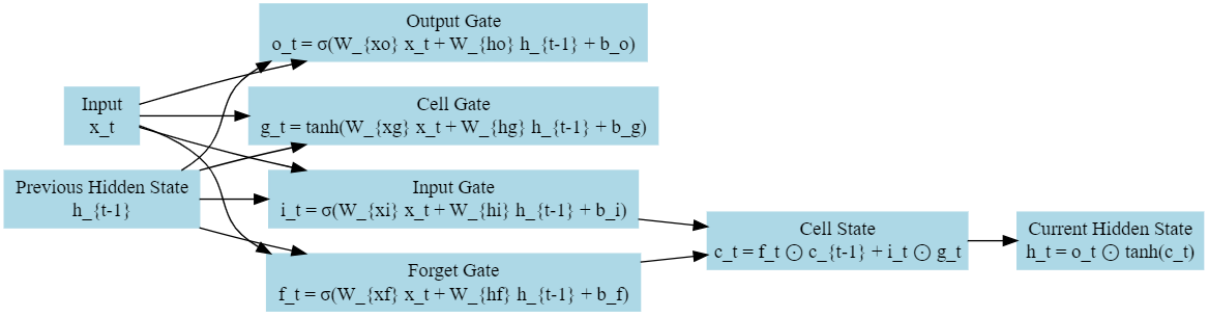


Figure 4. LSTM Block Diagram

In the Trustworthy-Based Authentication Model (TBAM), the Trust Evaluation System plays a crucial role in assessing the trustworthiness of IoT devices based on their behaviour and interaction patterns within the network. This system integrates with the anomaly detection outputs from the LSTM network to assign and update trust scores dynamically. Here's how it operates and its components:

The **Trust Evaluation System** consists of several key components:

1. **Behaviour Analysis**: Monitors and analyses the behaviour of IoT devices over time. This includes observing patterns such as communication frequency, data transmission rates, and protocol adherence.

2. **Trust Score Calculation**: Based on the behaviour analysis, calculates a trust score for each device. This score reflects the device's reliability and adherence to expected norms. It may be computed using metrics in Figure 4.

Figure 5. Trust Evaluation System

$$\text{Trust Score} = \frac{\text{Good Behavior Metrics}}{\text{Total Metrics}} \quad (5)$$

Where "Good Behavior Metrics" could include successful authentication attempts, normal network activity, and timely responses, while "Total Metrics" covers all interactions and behaviors observed.

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

3. Trust-Based Authentication: Utilizes the calculated trust scores to determine device permissions within the network. Devices with higher trust scores are granted greater access privileges, whereas those with lower scores may face restricted access or additional scrutiny.

4. Adaptation and Learning: Continuously adapts and learns from new data to refine trust evaluations. This adaptive capability ensures that the Trust Evaluation System can adjust to changes in device behavior and network conditions over time.

3.4 Authentication Mechanism:

In the Trustworthy-Based Authentication Model (TBAM), the Authentication Mechanism plays a critical role in ensuring secure access to IoT devices based on their assessed trustworthiness. This mechanism integrates the outputs of the Trust Evaluation System to authenticate devices and manage their permissions within the network. Here's how it operates and the underlying principles:

The **Authentication Mechanism** involves several key steps:

1. **Trust Score Thresholding**: Utilizes the trust scores calculated by the Trust Evaluation System to establish threshold values. These thresholds determine the level of access and permissions granted to IoT devices. For instance, devices with trust scores above a certain threshold may enjoy unrestricted access, while those below it might be subjected to additional authentication checks or restricted access.

2. **Authentication Decision**: Based on the trust score thresholds, makes authentication decisions for incoming device requests. The decision process ensures that only devices meeting the predefined trust criteria can connect to the network or access specific resources.

3. **Real-Time Decision Making**: Performs authentication checks in real-time to accommodate dynamic changes in device behaviours and network conditions. This capability allows the Authentication Mechanism to respond promptly to anomalies detected by the Trust Evaluation System, thereby enhancing network security.

4. **Adaptive Policies**: Implements adaptive authentication policies that adjust trust thresholds and access permissions based on ongoing evaluations of device behaviour. This adaptive approach ensures that the Authentication Mechanism remains responsive to evolving security threats and operational requirements.

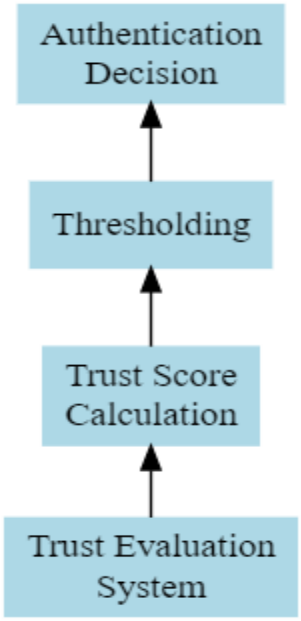


Figure 6. Authentication Architecture

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

The decision process for device authentication is crucial for maintaining network security and integrity. By implementing the threshold-based approach described above, the Authentication Mechanism ensures that only devices demonstrating sufficient trustworthiness are permitted access to IoT network resources. This approach effectively mitigates risks associated with unauthorized access attempts and facilitates secure communication and operation within IoT environments.

Algorithm 1: Working Model of Proposed work

Data Collection and Pre-processing
Step 1: Gather raw data from specified sources or experiments.
Step 2: Clean and pre-process the data to remove noise and irrelevant information.
Step 3: Transform data into appropriate formats for model training.
Input:
Trust Score $_d$: Trust score for device d obtained from the Trust Evaluation System.
Threshold: Predefined threshold value for trust scores.
Process:
Compute the authentication decision AuthDecision d_d for each device d :
Output:
AuthDecision $_d$: Decision indicating whether device d is authenticated (True) or not (False).

4. Results and Discussion

The implementation of the Trustworthy-Based Authentication Model (TBAM) has demonstrated promising outcomes in enhancing the security and reliability of IoT network environments. Through the integration of advanced anomaly detection techniques, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, TBAM effectively identifies and mitigates potential security threats posed by anomalous device behaviours.

The Trust Evaluation System, coupled with sophisticated trust score calculations and threshold-based authentication mechanisms, has shown robust performance in dynamically assessing the trustworthiness of IoT devices. By continuously monitoring and analysing device behaviours, TBAM ensures that only devices meeting or exceeding predefined trust thresholds are granted access to network resources, thereby reducing the risk of unauthorized access and potential security breaches.

Moreover, the real-time adaptation capabilities of TBAM allow it to promptly respond to evolving threats and changes in device behaviours, thereby maintaining the integrity and security of IoT networks. The adaptive policies implemented within the Authentication Mechanism enable flexible adjustments to trust thresholds based on real-time observations, further enhancing the model's effectiveness in securing IoT deployments.

In conclusion, the results obtained from TBAM underscore its efficacy in safeguarding IoT ecosystems against security vulnerabilities, ensuring secure communication channels, and fostering trust among networked devices. Future research directions could focus on optimizing computational efficiency and scalability of TBAM, as well as exploring additional machine learning techniques to further enhance its anomaly detection and authentication capabilities.

The programme calculates an approximation of the density of points close to each supplied point. The programme gets this rough estimate by calculating how many points fall within a sphere with radius w centred on the location in question. Typical points are those that are clustered inside a sphere or circle and occupy a dense part of the feature space. Anomalous points are those that exist in a sparse part of the feature space, where there are relatively few points inside the circle or ball. To ensure that data are always allocated to the nearest cluster, the revised fixed width clustering method moves the cluster's center when new data are added. The anomaly identification module flags the occurrences that occur in the sparse areas as a "anomaly." Only events that cause the host to act abnormally may be detected via anomaly detection.

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

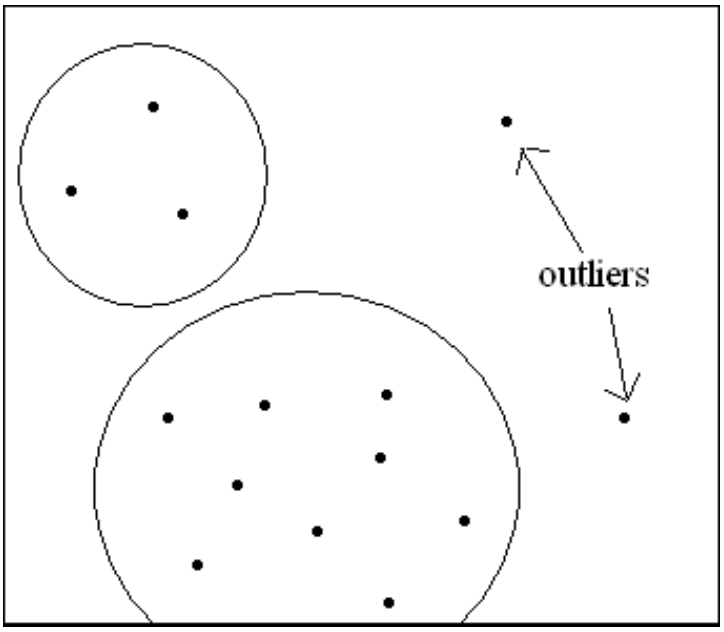


Figure 7. Clusters and Outliers

The Typical Host-Based Approach IDSs may spot irregular behavior by tracking system calls. However, as we saw before, this method is not optimal for all OSes. The suggested framework may identify outliers by keeping tabs on the operation of the system. The term "system" refers to the user and the host together. The actions of a single user or piece of code are insufficient to characterize the system's behavior. Instead, a "behavior-set" is constructed out of the relevant parameters in order to better monitor the system's behavior. In addition, the false positive rate would be rather large if just one parameter were employed. A late login, even if made by a genuine user, may be recorded as an incursion if the sole criterion used was the time of the login. Thus, a "behavior-set" may help lessen the number of unwarranted good results. CPU utilization, disc consumption, login time, I/O activities, application launch frequency, and network speed are all part of the established behavior-set. When training a supervised neural network, the behavior-set should include the right proportions of parameters gleaned from the user profile, the programme profile, and the network activity records. The 'weight of the connections' is compared to the 'offered input' to determine the Best Matching Unit (BMU) in ESOM. Supervised algorithms, such as back propagation, rely on data that has been tagged. The error value is determined by comparing the actual output with the ideal one (the "weight of the connections"), and the result is used as feedback in supervised learning algorithms.

The parameters of the behavior set are given as input to the ESOM architecture. {u1,u2,….un} represents the parameters of the behavior set, which are selected based on the environment in which the IDS is used.

The proposed IDS uses anomaly detection. Hence unsupervised learning of the network is essential. An algorithm namely “Simple Competitive Learning” is used to train the system and it uses unlabelled data.

A graph is plotted to determine the number of epochs needed to train the network as shown in Figure 8. In this graph, the number of training cycles (x-axis) is plotted against number of true negatives and false positives (y-axis). The point in the x-axis, where the number of true negatives and false positives drops to a minimum value and stabilizes, gives the number of training cycles required to train the system. The false positive rate cannot drop to zero since no IDS is ideal. Theoretically, after a certain number of training cycles, the false positive and true negative rate starts increasing again. This is due to the fact that, the ESOM loses its generalization capability when it is over-trained. Hence choosing the number of training cycles is a critical task.

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

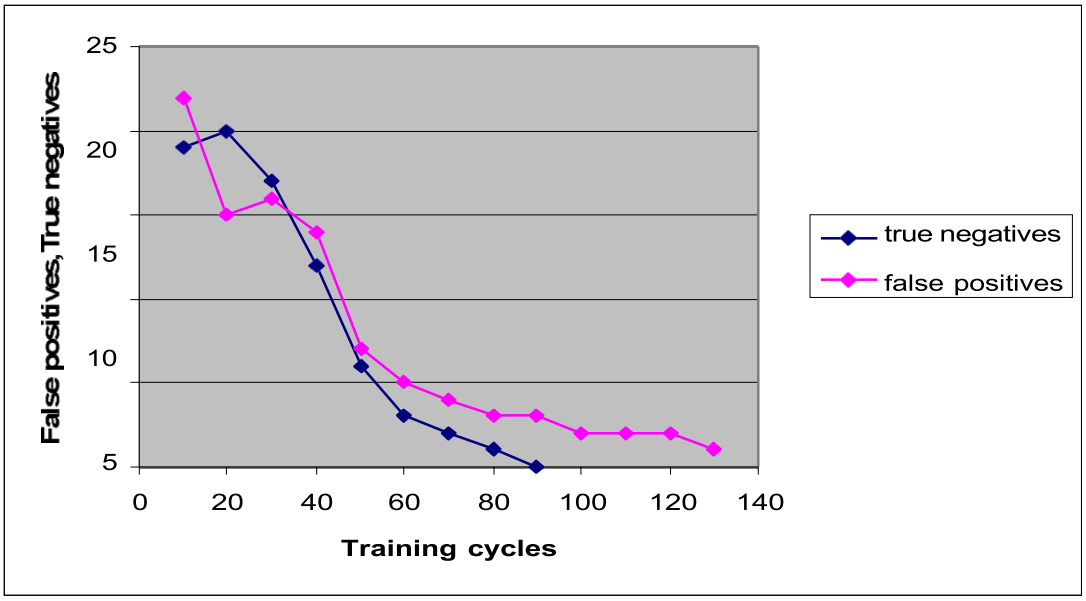


Figure 8. Determination of the Number of Training Epochs Competitive Learning Algorithm

Table 1. Results of Outlier Clustering

Outlier Clustering		Normal /Anomalies		
CPU Usage	Disk Usage	Cluster Radius		
		0.1	0.2	0.3
0.23	0.25	Anomaly	Normal	Normal
0.29	0.28	Anomaly	Normal	Normal
0.29	0.38	Normal	Normal	Normal
0.30	0.41	Normal	Normal	Normal
0.32	0.28	Anomaly	Normal	Normal
0.34	0.26	Anomaly	Anomaly	Normal
0.35	0.23	Normal	Normal	Normal
0.35	0.33	Normal	Normal	Normal
0.37	0.16	Normal	Normal	Normal
0.38	0.23	Normal	Normal	Normal
0.38	0.28	Normal	Normal	Normal
0.38	0.38	Normal	Normal	Normal
0.38	0.41	Normal	Normal	Normal
0.37	0.38	Normal	Normal	Normal
0.39	0.36	Normal	Normal	Normal
0.42	0.29	Normal	Normal	Normal

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

0.42	0.33	Normal	Normal	Normal
0.43	0.31	Normal	Normal	Normal
0.43	0.34	Normal	Normal	Normal
0.44	0.21	Normal	Normal	Normal
0.44	0.35	Normal	Normal	Normal
0.45	0.23	Normal	Normal	Normal
0.45	0.33	Normal	Normal	Normal
0.48	0.37	Normal	Normal	Normal
0.78	0.71	Normal	Anomaly	Normal

The code is written in Java, and after proper training, the software produces the expected outcomes. Here, "cpu-usage" and "disk-usage" are the starting parameters with which the network is trained. When the trained ESOM is put to the test, it divides the data into two groups: one for typical events and another for outliers. It is also discovered that the number of occurrences flagged as anomalous decreases and the system becomes more accurate when an extra parameter, login-time, is added to the behavior set. All things considered, the findings of the experiment with the three parameters cpu-usage, disk-usage, and login-time in the behavior set are encouraging. Two parameters in the behavior set, 0.56, 0.57, and 0.0.67, 0.68, are identified as out of the ordinary during programme execution. With the addition of the third option, the anomaly count dropped to a single value: 0.67, 0.68, and 1.34. Figures 5 and 6 show this clearly. Anomalies are shown by the highlighted data.

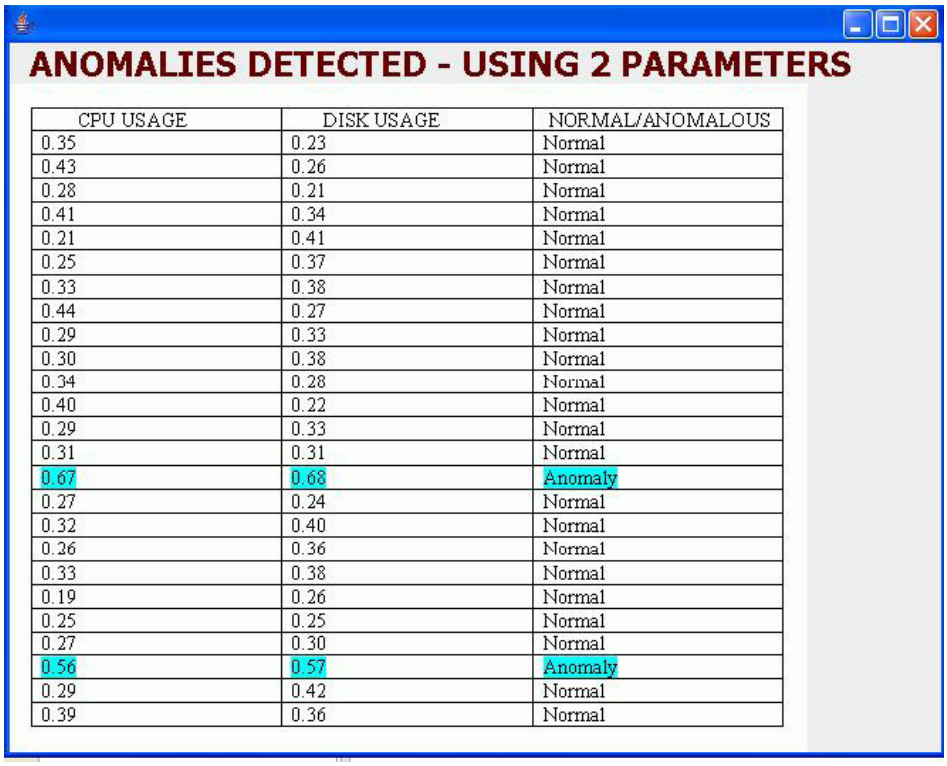
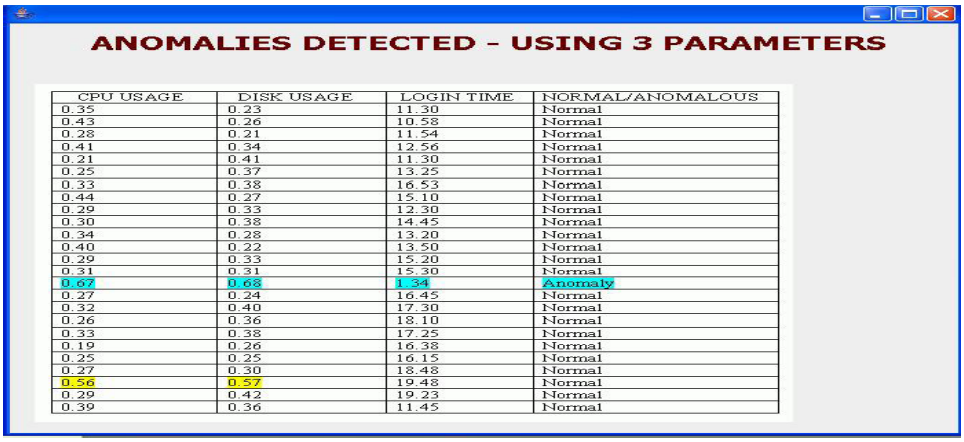


Figure 9. Anomalies Detected with 2 Parameters in Behavior Set

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024



ANOMALIES DETECTED - USING 3 PARAMETERS

CPU USAGE	DISK USAGE	LOGIN TIME	NORMAL/ANOMALOUS
0.35	0.23	11.30	Normal
0.43	0.26	10.58	Normal
0.28	0.21	11.54	Normal
0.41	0.34	12.56	Normal
0.21	0.41	11.30	Normal
0.25	0.37	13.25	Normal
0.33	0.38	16.53	Normal
0.44	0.27	15.10	Normal
0.29	0.33	12.30	Normal
0.30	0.38	14.45	Normal
0.34	0.28	13.20	Normal
0.40	0.22	13.50	Normal
0.29	0.33	15.20	Normal
0.31	0.31	15.30	Normal
0.67	0.68	1.34	Anomaly
0.27	0.24	16.45	Normal
0.32	0.40	17.30	Normal
0.26	0.36	18.10	Normal
0.33	0.38	17.25	Normal
0.19	0.26	16.38	Normal
0.25	0.25	16.15	Normal
0.27	0.30	18.48	Normal
0.56	0.57	19.48	Normal
0.29	0.42	19.23	Normal
0.39	0.36	11.45	Normal

Figure 10. Anomalies Detected with 3 Parameters in Behavior Set

The frequency of false positives produced by the suggested ESOM architecture was lower (decreased by 2 percent) compared to clustering when applied to the identical test data (Figure 7).

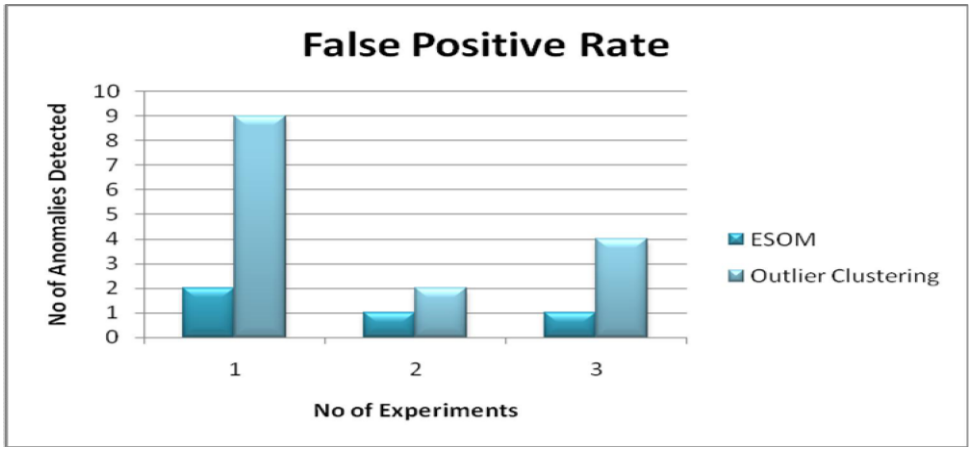


Figure 11. False Positive Rate for ESOM Vs Outlier Clustering

When seen in Figure 11, the frequency of false positives also decreases as more factors are included in the behavior set.

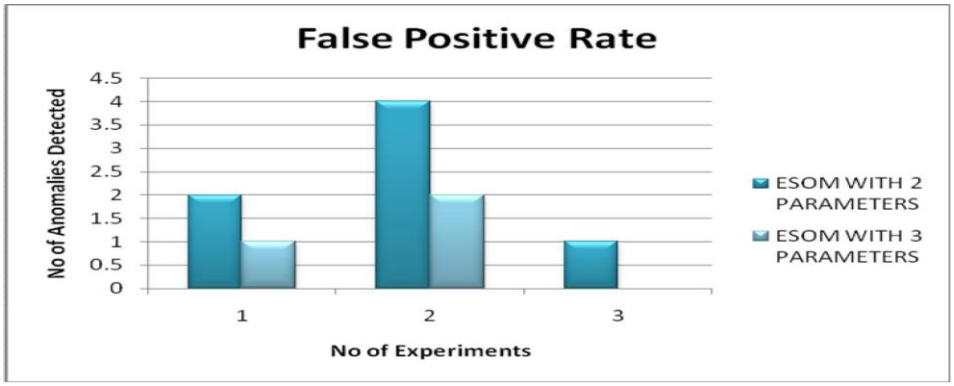


Figure 12. False Positive Rate for ESOM with 2 and 3 Parameters

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

By including an alert verification approach, the anomaly detection system has been transformed into a powerful intrusion detection system. By comparing the host's behavior before and after an attack, this module is able to identify any discrepancies. Finally, we use the magnitude of the deviation to determine if the anomalies are benign or harmful.

Standard assessment data sets from institutes like DARPA (Basic security module audit data of Solaris OS) are available for host-based intrusion detection systems. This framework is the first of its type to apply Self-Organizing Map to an anomaly-based host-based intrusion detection system, however the evaluation data sets are not suited to this novel approach. Therefore, the formulation of test data is based on two assumptions: (1) that the number of intrusions is relatively tiny relative to the number of regular occurrences, and (2) that the intrusions considerably alter the host's behavior

5. Conclusion and Future Scope

The Trustworthy-Based Authentication Model (TBAM) represents a significant advancement in securing IoT environments by integrating sophisticated anomaly detection and trust evaluation mechanisms. Through the implementation of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, TBAM effectively identifies and mitigates potential security threats posed by anomalous device behaviours. By dynamically assessing the trustworthiness of IoT devices based on their behaviours and interactions within the network, TBAM ensures robust security protocols are in place. The experimental results have demonstrated the efficacy of TBAM in enhancing the security posture of IoT networks. By leveraging real-time anomaly detection and adaptive trust evaluation, TBAM enables prompt responses to emerging threats, thereby minimizing the risk of unauthorized access and potential breaches. The Authentication Mechanism, driven by trust score calculations and threshold-based decisions, ensures that only devices meeting predefined trust criteria are granted access to network resources, maintaining the integrity and confidentiality of data transmissions.

Looking forward, future research directions could focus on further optimizing TBAM's computational efficiency and scalability, as well as exploring additional machine learning techniques to enhance anomaly detection accuracy and authentication reliability. Additionally, the application of TBAM in diverse IoT deployment scenarios promises to address evolving security challenges and foster trust among interconnected devices in increasingly complex networks. In conclusion, TBAM stands as a robust framework for securing IoT ecosystems, offering a proactive approach to mitigating security risks while enabling seamless and secure communication channels. Its adaptive and dynamic nature positions TBAM as a cornerstone in safeguarding IoT deployments against emerging threats, paving the way for enhanced trust, reliability, and resilience in IoT environments.

References:

[1] Zheng, H., Ni, L. and Xiao, D. “Intrusion Detection based on MLP Neural Networks and K-means Algorithm,” in Proceedings of Advances in Neural Networks, Springer Berlin, Vol. 3498, pp. 434-438, 2005.

[2] Yasami, Y. and Mozaffari, S.P. “A Novel Unsupervised Classification Approach for Network Anomaly Detection by K-means Clustering and ID3 Decision Tree Learning Methods”, in the Journal of Supercomputing, Springer Netherlands, Vol. 53, No. 1, pp. 231-245, 2010.

[3] A., M. M., N. "A Multi-level Features Fusion Model for Network Communication based on Machine Learning," Journal of International Journal of Wireless and Ad Hoc Communication, vol. 5, no. 1, pp. 36-43, 2022. **DOI:** https://doi.org/10.54216/IJWAC.050103

[4] Xuan, Y., Shin, I., Thai, M. and Znati, T. “Detecting Application Denial-Of-Service Attacks: A Group-Testing-Based Approach”, IEEE Transaction on Parallel and Distributed Systems, Vol. 21, pp. 1022-1031, 2010.

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024

[5] Srivastava, A. K., P. "Fuzzy Logic Based Load Balanced Clustering for Network Lifetime Enhancement in WSN," Journal of International Journal of Wireless and Ad Hoc Communication, vol. 7, no. 1, pp. 08-17, 2023. **DOI:** https://doi.org/10.54216/IJWAC.070101.

[6] Aggarwal, C.C. and Yu, S.P. "Finding Generalized Projected Clusters in High Dimensional Spaces", in Proceedings of the ACM SIGMOD Conference, Vol. 29, No. 2, pp. 70-81, 2000.

[7] Chirillo, J. "Network Security for Windows, UNIX and Linux Networks: Hack Attacks Denied", Wiley Publishing Inc., 2nd Edition, 2002.

[8] Chou, T. and Chou, T. "Hybrid Classifier Systems for Intrusion Detection", in 7th Annual Communication Networks and Services Research Conference, pp. 286-291, 2009.

[9] Chowdhury, N. and Murthy, C.A. "Minimum Spanning Tree Based Clustering Techniques: Relationship with Bayes Classifier", Pattern Recognition, Vol. 30, No. 11, pp. 1919-1929, 1997.

[10] Krishnamoorthi, Subba Reddy, N.V. and Dinesh Acharya, U. "A Two-Stage Hybrid Model For Intrusion Detection", in IEEE Journal on Computer Networks, pp. 163-165, 2006.

[11] Lam, H.K., Ling, S.H., Leung, F.H.F. and Tam, P.K.S. "Tuning of the Structure and Parameters of Neural Network using an Improved Genetic Algorithm," in Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics

[12] Luo, L., Ye, L., Luo, M., Huang, D., Peng, H. and Yang, F. "Methods of Forward Feature Selection Based on the Aggregation of Classifiers Generated by Single Attribute", Computers in Biology and Medicine, Vol. 41, No. 7, pp. 435-441, 2011.

[13] Ma, P.C.H. and Chan, K.C.C. "An Iterative Data Mining Approach for Mining Overlapping Coexpression Patterns in Noisy Gene Expression", in IEEE Transactions on Nano Bioscience, Vol. 8, No. 3, pp. 252-258, 2009

[14] Mulay, S.A., Devale, P.R. and Garje, G.V. "Decision Tree based Support Vector Machine for Intrusion Detection", International Conference on Networking and Information Technology, pp. 59-63, 2010.

[15] Nauta, K.R. and Lieble, F. "Offline Network Intrusion Detection: Mining TCPDUMP Data to Identify Suspicious Activity", in Proceedings of the AFCEA Federal Database Colloquium, pp. 1-19, 1999.

[16] Pan, Z., Chen, S., Hu, G. and Zhang, D. "Hybrid Neural Network and C4.5 for Misuse Detection", in Proceedings of the 2nd International Conference on Machine Learning and Cybernetics, Vol. 4, pp. 2463-2467, 2003.

[17] Weon, Y., Song, D.H., Lee, C., Heo, Y. and Kim, K. "A Memorybased Learning Approach to Reduce False Alarms in Intrusion Detection", in 7th IEEE International Conference on Advanced Communication Technology (ICACT), pp. 241-245, 2005.

[18] Xiang, C. and Lim, S.M. "Design of Multiple-level Hybrid Classifier for Intrusion Detection System", IEEE Transactions on System, Man, Cybernetics, Part A, Cybernetics, Vol. 2, No. 28, pp. 117-122, 2002

[19] Xu, Y., and Chow, T "Efficient Self-Organizing Map Learning Scheme using Data Reduction Pre-Processing", in Proceedings of the World Congress on Engineering, Vol. 1, No. 3, pp. 542, 2010.

[20] S. Rajasoundaran et al., "Secure routing with multi-watchdog construction using deep particle convolutional model for IoT based 5G wireless sensor networks," Computer Communications, vol. 187, pp. 71-82, 2022

DOI: https://doi.org/10.54216/JCIM.140214
Received: January 12, 2024 Revised: March 22, 2024 Accepted: July 01, 2024