

Neural Representation Models for Intelligent Threat Identification in IoT Networks

¹Thanuja Narasimhamurthy,²Gunavathi Hosahalli Swamy

¹Assistant Professor, Department of Computer Science & Engineering,
Bangalore Institute of Technology, Bengaluru, Affiliated to Visvesvaraya Technological University,
Belagavi-590018, Karnataka, India

²Associate Professor, Dept. of Computer Science & Engineering,
Bangalore Institute of Technology, Bengaluru, Affiliated to Visvesvaraya Technological University,
Belagavi-590018, Karnataka, India

DOI: <https://doic.org/10.10399/JBSE.2025458054>

Article History: **Received: January 2025** **Revised: April 2025** **Accepted: July 2025**

Abstract: - Machine Learning is one of the potential solutions towards optimizing the security strength towards identifying complex forms of threats in Internet-of-Things (IoT). However, a review of existing machine learning-based approaches showcases their sub-optimal performance when exposed to dynamic forms of unseen threats without any apriori information during the training stage. Hence, this manuscript presents a novel machine-learning framework towards potential threat detection capable of identifying the underlying patterns of rapidly evolving threats. The proposed system uses a neural network-based learning model emphasizing representation learning where an explicit masked indexing mechanism is presented for high-level security against unknown and dynamic adversaries. The benchmarked outcome of the study shows to accomplish 11% maximized threat detection accuracy and 33% minimized algorithm processing time.

Keywords: - Internet-of-Things, Machine Learning, Dynamic Threats, Neural Network, Performance Index.

1. Introduction

Internet-of-Things (IoT) comprises various forms of software and hardware connected in a larger context to disseminate and analyze data from various application perspectives [1]. Irrespective of various related concerns, security issues dominate the research communities, even if a more extensive set of security solutions is being introduced for IoT [2]. From the perspective of IoT security, various challenges are associated with performing frequent sets of operations, e.g., data encryption, access control, firmware updates, and device authentication [3]. Some key security issues in IoT are mainly associated with scalability, limited resources, diverse ecosystems, privacy concerns, and physical security [4]. There is an evolving landscape of cyber threats in IoT, which are consistently being researched [5].

Out of various security solutions, Artificial Intelligence (AI) has eventually played a robust solution to deal with this complex security problem with various evolving machine learning algorithms [6]-[9]. Existing studies have noted the adoption of AI in anomaly detection [10], adaptive security [11], automated threat response [12], and predictive capabilities [13]. However, there are rising concerns about adopting machine learning algorithms in IoT security, too. The primary limitation is associated with the dependency of massive data volumes for machine learning models which is a rising concern towards sensitivity and data privacy. This problem is also linked to the increasing requirement for training data of higher quality and model complexity. Further, there is always the risk of overfitting thereby restricting the capability of the algorithm to adapt for fluctuating data.

However, one notable problem that is normally overseen by the majority of existing studies is that once the dataset is trained, the learning model completely depends on it for its predictive analysis.

Under such a scenario, if there is any form of deviation in the network scenario or if there is any form of unknown or dynamic threats, the trained model fails to support the higher and more reliable threat detection by the existing machine learning model. Hence, existing studies using machine learning may sound good, but in reality, their applicability is restricted to static environment implementation only.

Hence, the prime gap of the existing study is mainly related to the non-updating process for the trained model during the actual deployment process by existing machine learning models. Further, there is a need to develop such a model that can further address the overfitting and reliability problems while using machine learning in order to balance the demands of successful, updated, and reliable threat detection with a cost-effective learning model. The related *background work* of literature shows that there are various contributions of machine learning-based approaches towards boosting the security features in IoT [14]. There are various studies reviewed considering adopted methodologies, features, and datasets. Table 1 highlights the characteristic being obtained from existing studies [15]-[30]. Hence, it can be seen that XGBoost, Support Vector Machine (SVM), Random Forest (RF), and Decision Tree (DT) are frequently adopted approaches noted in existing approaches, which is shown in the table 1.

Table 1. Review of Existing Studies

Studies	Methods	Features	Dataset
[15]	Multilayer Perceptron, SVM, RF	Health-related	WUSTL-EHMC2020, ICU-IOMT
[16]	DT	18 sensor-related features	WSN-DS
[17]	Collaborative Learning	Traffic data	ECU-IoHT
[18]	XGBoost, Fisher score model, genetic algorithm	115 features from Traffic data	N-BaIoT
[19]	Ensemble	4 Features from network traffic	NSL-KDD
[20]	CNN, nature-inspired algorithm	34 features from network traffic	IoT-MQTT
[21]	XGBoost, RF	115 features from Traffic data	N-BaIoT
[22] [23]	SVM	Data features related to activity, 80-traffic feature	Synthetic, CSE-CIC-IDS2018
[24]	SVM(multi-class), DT	20 features from IoT traffic	Multistep Cyber-Attack Dataset
[25]	SVM, K-Nearest Neighbour, RF	A large number of features	Multiple datasets
[26]	CNN	Features for 25 classes	MaleVis
[27]	Machine learning, encryption	Features related to traffic flow	Synthetic dataset
[28]	multiple machine learning	Traffic-related feature	IoT-23
[29]	multiple machine learning	45-features	TON-IoT
[30]	multiple machine learning	6-features related to traffic flow	NFUQ-NIDS-v2

The *research problems* associated with existing studies are manifold. With the advancement of varied networking in large scale and distributed manner, there is also an immense increase in different threats. Such threats have been reported to have high potential and can bypass any security monitoring system in IoT (e.g. firewall). The existing security firewall system can only resist the threats already considered while developing the security patch for the firewall. Hence, firewall systems cannot identify any threats that are not declared and defined well in their standard database system. Hence,

there is a need for an innovative system capable of identifying a security threat of unknown form different from that used in developing a firewall system.

Therefore, the *proposed study aims to develop a computational model* capable of harnessing the potential of a machine-learning approach towards dynamic, unknown, and uncertain threat detection in IoT. The *value-added contributions* in current work are as follows:

- i) A novel threat detection scheme has been presented capable of learning the dynamic patterns and correlation of potential vulnerabilities,
- ii) A distinct updating scheme has been introduced that can optimize the data quality of training model to realize the dynamic events in incoming traffic,
- iii) A simplistic and yet innovative neural network based machine learning model has been presented that can perform representation learning to extract more significant underlying patterns of threats,
- iv) A unique masked indexing mechanism has been presented towards safeguarding the labels of trained data from adversaries.

The paper's organization is portrayed as shown here. A brief introduction was given in the Section 1 in the previous paragraphs. The Section 2 presents a discussion of methods, while the result is discussed in Section 3, and the conclusion is presented in Section-4. This is followed by exhaustive set of reference papers & the author biographies.

2. Methodology Adopted

The proposed system aims to develop a novel detection scheme of potential threats by harnessing machine learning towards securing future communication systems i.e. IoT. An IoT infrastructure is formed by considering a specific number of IoT devices that capture the data and forward it to a processing unit hosted within an edge device. The processing unit, considering an additional 50⁺ features using a standard dataset related to traffic flow, further executes the algorithms towards scrutinizing threats that are hosted in a connected cloud environment. Analytical modelling is considered for developing the research methodology, where the complete operation is classified into dual implementation stages. The first implementation is towards identifying the potential threat, while the second implementation is about applying a neural network based machine learning model to perform updating of incoming threats. The Fig. 1 highlights the adopted architecture of the proposed methodology.

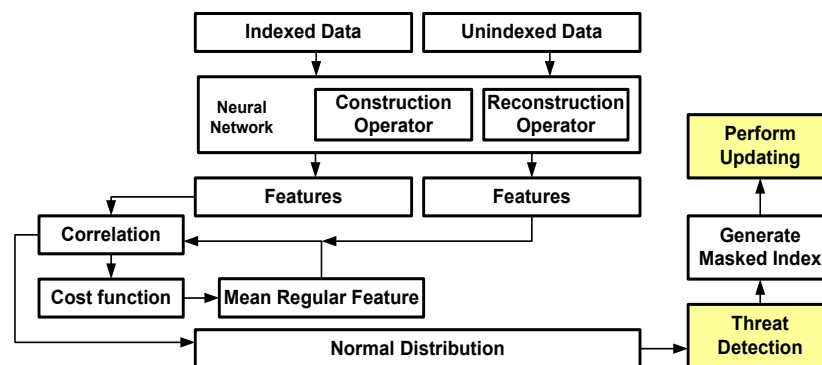


Fig. 1. Architecture of Proposed Research Methodology

According to Fig.1, the traffic information is split into indexed and unindexed data which is subjected to a neural network model with an inclusion of construction and reconstruction operator. The neural network model adopted towards performing learning of traffic data representation where the input traffic data is initially subjected to forward pass where construction operator is used to generate a latent representation while reconstruction operator is used to generate reconstruction of original input using latent representation generated in prior stage. The operation yields features that are subjected to correlation analysis followed by using cost function with an objective to make the reconstruction matching with the original input. The cost function adopted in the proposed system is basically designed for learning representation that is capable of extracting both similarities and differences among data points. This is done with the objective to incentivise regular traffic and penalize malicious traffic. The novelty of this neural network model is that it uses its operator and cost function to reduce the distance between the positive traffic data (representing regular traffic) and increasing the distance between dissimilar data (representing malicious traffic).

The operations of the cost function result in mean regular features, which, along with extracted values of normal distribution, are subjected towards assessment of regular and malicious traffic, thereby accomplishing the threat detection operation. After the threat detection is performed, the next task is to increase the adaptability of the model to perform detection over a stream of dynamic traffic information. The same neural network model is trained with a random weight, resulting in the masked index which represents a second layer of encoded value of labelled data of traffic. The beneficial aspect of this operation is that the adoption of masked label offers further data integrity as well as data privacy as no attacker will ever be able to compute the actual label information of the traffic data. Another beneficial aspect of this module is that – the model can now be able to easily identify any form of spurious traffic if an attacker attempts to tamper with either the data or the label of it.

The tampered label of the data by the attacker is less likely to match with the computed value of the masked index presented in the proposed system and hence strongly acts similarly to a trapdoor function used in a conventional cryptography-based approach. Further, the proposed study introduces updating operations in order to increase the capability of coverage of trained data by the proposed neural network model to be able to resist potentially dynamic forms of threats. Interestingly, the internal operation of the scheme is quite progressive and less iterative. Eventually, it uses machine learning methodology which contributes towards an effective balance between security features and computational features. An illustration of design implementation follows next:

2.1 Potential Threat Identification

This is the first module of implementation that is responsible for threat identification. The prime task of this algorithm is basically to perform the identification of potential threats, followed by the generation of a masked index. Basically, the term masked index is meant for safeguarding the labels assigned for the incoming traffic data, which otherwise could be tampered with or eavesdropped by an attacker. The steps of the proposed algorithm towards threat identification are as follows:

Algorithm for Threat Identification

Input: T

Output: TI

Start

1. init $T \rightarrow (T_r, T_m)$

2. $(T_r, T_m) \rightarrow [(\alpha_r, \beta_r), (\alpha_m, \beta_m)]$
3. compute $\rho = f_1(tr_i, tr_j)$
4. $\lambda = f_2(\rho, \delta)$
5. $\lambda_1 = \text{argmin}(\Sigma \lambda)$
6. $TI = f_3\{\rho(r, m)\}$

End

The illustration of the algorithm is as follows: The algorithm takes the input of T (traffic data), which, upon processing, will lead to the generation of TI (Threat identification). The algorithm initializes the traffic data T concerning regular traffic T_r and malicious traffic T_m (Line-1). Further, the traffic data is represented in the form of input traffic vector (α_r, α_m) of regular and malicious forms as well as associated indexes (β_r, β_m) of regular and malicious forms, respectively (Line-2). It will eventually mean that the traffic data T will consist of input traffic vector α of s size while each index β is initialized with a binary range of $[0,1]$ where 0 and 1 depict regular and malicious traffic, respectively. The proposed algorithm considers the trained representation associated with each input traffic data as tr (\in) TR, while the proposed neural network model used for threat detection is represented by π_w , where w represents the weight associated with the neural model.

The proposed system uses a unique cost function that targets to increase the correlation between regular traffic $(tr_{r,i}, tr_{r,j})$ and decrease the same for reference sample $tr_{r,i}$ and malicious traffic $tr_{m,k}$. The variable $tr_{r,i}$ represents the reference sample associated with regular traffic information while the value of the subscripts (i, j) is $(1, 2, 3, \dots, I_r)$, where variable I_r represents the index for regular traffic. The value of subscript k is $(1, 2, 3, \dots, I_m)$, where variable I_m represents the index for malicious traffic. In the consecutive part of algorithm implementation, the scheme computes correlation attribute ρ by constructing an explicit function $f_1(x)$ considering input arguments of regular traffic information tr_i and tr_j (Line-3). The function $f_1(x)$, which is intended to evaluate the similarity between pairs of traffic data points, is used by the system to calculate the correlation attribute ρ . In particular, $f_1(x)$ computes the dot product of two traffic samples' feature vectors and normalizes the outcome using the Euclidean distance. This process makes sure that the correlation accurately represents how similar the samples are, which is essential for differentiating malicious traffic from legitimate traffic. The neural network learns to distinguish between malicious and legitimate traffic by using the correlation score as input to the cost function λ . The computation of this explicit function $f_1(x)$ is empirically expressed as follows

$$f_1(x) = (tr_i \cdot tr_j) \cdot [abs(tr_i, tr_j)]^{-1} \tag{1}$$

After the explicit function $f_1(x)$ computation, the system computes the value of the correlation attribute ρ , which is further utilized towards its consecutive steps for computing cost function λ . A closer look into Line 4 showcases that the computation of cost function λ is carried out by developing another explicit function $f_2(x)$ with an input argument of correlation attribute ρ and adjustment parameter δ . The step-wise computation of cost function λ is shown as follows:

$$\lambda_{i,j} = c \cdot \frac{A_1}{A_2} \tag{2}$$

$$\lambda = [I]^{-1} \sum \lambda_{i,j} \tag{3}$$

A closer look into the above expressions shows that the proposed scheme initially computes expression (2), which is then used in expression (3). In the initial computational step of expression

(2), the variable A_1 represents an exponential form of correlation attribute ρ for (i, j) divided by adjustment parameter δ . At the same time, the variable A_2 represents the summation of the newly accomplished value of A_1 and all additive values of A_1 with the lower range of (i, k) and index range of (I_r, I_m) . In the next computational step of expression (3), the variable I represents $I_r(I_r - 1)$, where I_r is the index for regular traffic and the second component represents all additive values of expression (2).

The algorithm minimizes the cost function associated with both construction co and reconstruction re to efficiently learn the traffic representation to obtain the new value of cost function λ_1 (Line-5). The variable $\Sigma\lambda$ represents the summation of λ_{co} and λ_{re} . In the final step of the algorithm (Line-6), another explicit method $f_3(x)$ is used for Higher Probability Assessment (HPA) in order to derive an accurate correlation score ρ with the normal distribution of traffic with respect to both regular and malicious. This information of the identified threat is stored in matrix TI .

2.2 Updating Operation for Incoming Threat

This is the second part of implementation which is mainly concentrated towards performing updated operations for all incoming threat-prone traffic. A closer look into the adoption of machine learning approaches towards threat detection in existing systems is always found to depend on its predictive operation on its trained data. The very assumption that all trained data will consist of complete possibilities of traffic events upon exposing them in a deployment scenario is impractical, especially regarding dynamic threats. Hence, the prime objective of this algorithm is to increase the adaptability of the threat detection model to understand and realize the dynamicity involved in the traffic environment by generating updated information on incoming threats. The core contribution of this model is its freedom from any form of demands of manual indexing tasks while performing training operations over streams of traffic information. The algorithmic steps are as shown below:

Algorithm for Updating Incoming Threat

Input: T_o, w

Output: up

Start

1. *init* w
2. *train param*
3. $\alpha, \beta \leftarrow \alpha_o, \beta_o$
4. **For** $i = \alpha_i$
5. $N[(co, re)(r, m)] = f_4(param_1)$
6. **For** $j = 0 : (h - 1)$ **do**
7. $res(co, re)_{i+j} = \pi_w(\alpha_{i+j})$
8. $\beta_1 = f_5(param_2)$
9. $up(\alpha, \beta) = f_6[(\alpha, \beta), (\alpha_{i+j}, \beta_{i+j})]$
10. **End**
11. **End**
12. *initiate training(param₂)*

End

The above-shown algorithm takes the input of T_o (dataset for Training) and w (weight), which, after processing, yields an outcome of up (updated Information of Threat). Neural network framework π_w is initialized considering weights w of arbitrary form (Line-1) is used for training the system considering a pilot sample of indexed traffic information that consists of input traffic vector α_o and associated indexes β_o considered for pilot epoch round e_o (Line-1 and Line-2). The variable $param$ represents input traffic vector α_o , associated indexes β_o , and epoch e_o (Line-2). The input traffic vector α_o and associated indexes β_o are further assigned to the new mapping attribute of input traffic vector α and associated indexes β (Line-3).

The following line of operation considers all the input traffic vector α_i (Line-4), followed by an evaluation of the normal distribution N (Line-5). For this purpose, an explicit function $f_4(x)$ is constructed to assess the suitability of normal distribution considering $param_1$ representing pilot input traffic vector α_o , input traffic vector α , and Neural network framework π_w (Line-5). The evaluated normal distribution suitability score is then assigned to matrix N considering construction operator co , reconstruction operator re associated with regular traffic r and malicious traffic m (Line-5). In simple words, this operation (Line-5) extracts the normal distribution of construction operator co and reconstructor operator re using pilot traffic information T_o .

Further, the algorithm generates the masked index (Line-5 to Line-8) followed by the generation of updated information (Line-9 to Line-12). Considering the phases involved in Training ranging from 0 to $(h - 1)$ as shown in Line-6, the algorithm obtains results res concerning construction operator co and reconstruction operator re (Line-7). It eventually infers towards accomplishing results concerning input traffic vector α (Line-7). In the consecutive process, the algorithm allocates the masked index to all the recently generated input traffic α_j that has already been detected in the first algorithm using the Neural network framework π_w . The function $f_5(x)$ is meant for undertaking a decision considering newly generated result res in the previous step concerning $param_2$ (Line-8) that can be further empirically expressed as follows:

$$\beta(co)_{i+j} \rightarrow res(co), N(co, r), N(co, m) \tag{4}$$

$$\beta(re)_{i+j} \rightarrow res(re), N(re, r), N(re, m) \tag{5}$$

In the above expression (2) and (3), the $\{res(co), N(co, r), N(co, m)\}$ and $\{res(re), N(re, r), N(re, m)\}$ together constitute of $param_2$. The function $f_5(x)$ generates a value of masking index β as 1 if it finds error-prone input followed by a flagging alert to the system. It will also mean that the previous algorithm plays the role of inference to yield outcomes in this model while performing generation for masked indexes. Only the outcome of masked index β_{i+j} that is found to have a higher score is considered further, followed by randomly permuting masked index β . This algorithm's final step is updating both input traffic vector α and associated masked indexes β (Line-9). For this purpose, function $f_6(x)$ has been constructed, which is mainly performing union operations between α , α_{i+j} , and β_{i+j} while the Training is carried out on $param_2$, which consists of α , β , π_w , and new epoch e_1 (Line-12). It can be seen that training dataset T is now expanded with the inclusion of a new input traffic vector α_j and its associated masked index β .

Considering h number of incoming traffic being newly identified, new epoch e_1 is used for adjusting the neural network model π_w , which eventually means that the final index consists of both ground-truth information index and masked index. The following section discusses about accomplished results. By doing away with manual labeling and ongoing retraining, the Masked Indexing Mechanism presents a revolutionary method for dynamic threat detection. By employing

“*masked indexes*” to conceal traffic labels and thwart label manipulation by adversaries, it enables the system to automatically identify and adjust to changing threats in real-time. Even in dynamic, label-free situations, which are typical of IoT networks, this approach guarantees reliable threat detection. Because iterative re-training of the system is not necessary, it also reduces computing overhead, increasing efficiency. Because of the increased accuracy, shorter processing times, and more trustworthy threat detection that results, it is especially useful in settings where data patterns are ever-changing.

3. Results

The scripting of the proposed study is carried out on a python environment on a normal 64-bit Windows environment considering two standard benchmarked datasets implemented on a standard 64-bit Windows system with NVIDIA GEFORCE GTX with 16GB RAM and Intel Core i5 processor. The environment was retained the same for assessing the proposed existing system. The first dataset is the NSL-KDD dataset [31], where 25 classes of malicious forms are considered for training operations encapsulating attackers that attempt to explore various types of attacks as follows ...

- i) *Attacker Type-1*: This type of attackers exploit the weakness of network structure, e.g., Port scanning, ping sweeping, network mapping, service enumeration, vulnerability scanning.
- ii) *Attacker Type-2*: This type of cyber attack to gain illegitimate access to control the network or machine of the victim node remotely, e.g., SQL injection, buffer overflow, remote code execution.
- iii) *Attacker Type-3*: This type of cyber attack is used for gaining access to root administrative accounts illegally, e.g., Kernel exploits, misconfiguration of Set Group ID (SGID) or Set User ID (SUID).
- iv) *Attacker Type-4*: This attacker is meant to flood illegitimate requests in order to disrupt the normal services from a server, e.g., Volume-based attacks, protocol attacks, application layer attacks, Denial-of-Service (DoS), and distributed DoS (DDoS).

The second dataset is the UNSW-NB15 dataset [32], with more than 100GB of raw traffic information extracted from artificially constructed adversaries with 9 classes of attack with 49 features. This dataset provides vulnerable environmental assessment for DoS, worms, malware, etc. There are a total of 175, 341 and 82, 332 records used for Training and testing. The proposed scheme uses 5 layers with following sizes

- i) For first dataset (size of input layer = 121, size of constructor layer = 64, size of hidden layer = 32, size of reconstructor layer = 64, size of output layer = 121), and
- ii) For second dataset (size of input layer = 196, size of constructor layer = 128, size of hidden layer = 64, size of reconstructor layer = 128, size of output layer = 196).

The adjustment parameter δ is set to 0.02 with 128 batch size, while 0.001 is set for the rate of learning considering rectilinear Unit (ReLU) as the activation function. The scheme uses gradient descent of stochastic form to play the role of optimization in the proposed machine learning model. The NSL-KDD dataset consists of network-based attacks, while the UNSW-NB-15 dataset consists of more diverse attack coverage on more realistic scenarios of IoT. Adoption of these datasets facilitates to simulation of general attacks based on network vulnerabilities in IoT.

The outcome is assessed for varied accuracy parameters (precision, recall, and F1-score), overall accuracy, and algorithm processing time. The benchmarking is carried out by comparing the proposed

system with existing machine learning algorithms that are frequently reported in the literature, e.g., XGBoost, SVM, RF, and DT (Fig. 2 to Fig. 6). The outcome of the analysis eventually shows the proposed scheme of a neural network to perform better performance in contrast to existing machine learning approaches with respect to accuracy and processing time-based evaluation metric.

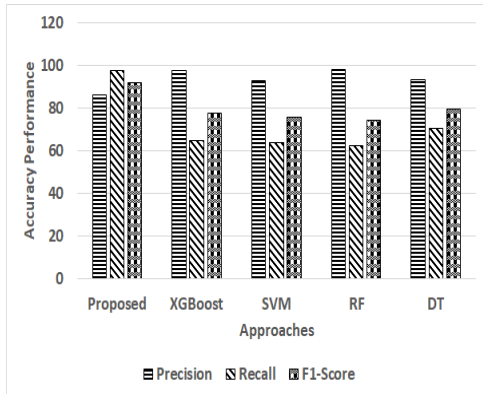


Fig. 2. Accuracy for NSL-KDD Dataset

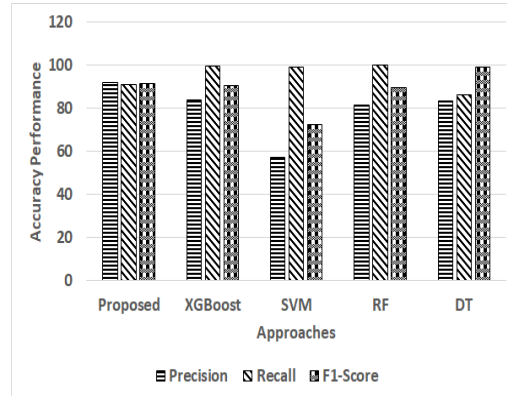


Fig. 3. Accuracy for UNSW-NB15 Dataset

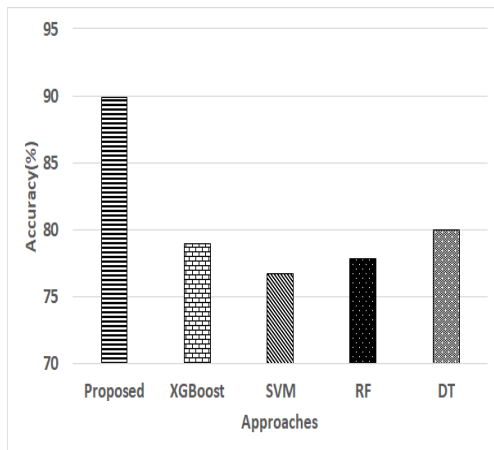


Fig. 4. Accuracy Score for NSL-KDD Dataset

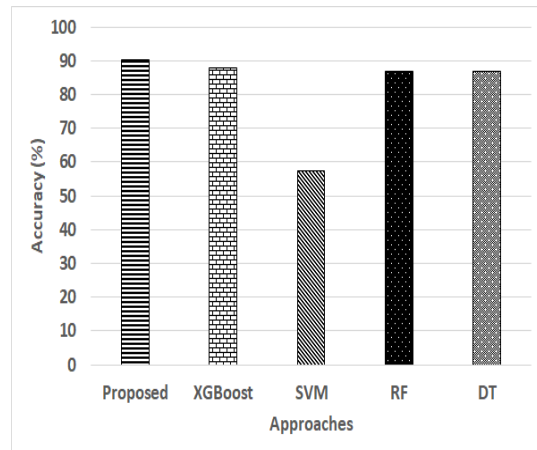


Fig. 5. Accuracy Score for UNSW-NB15 Dataset

3.1 Discussion of Accuracy Outcomes

From the (Fig. 2 to Fig. 5), it could be inferred that the quantification of accuracy outcomes shows that there is no significant difference in accuracy outcomes for the first and second datasets. The proposed system is found to offer approximately 12% and 10% better accuracy scores recorded for the first and second datasets, respectively, in contrast to existing machine learning approaches. The justification is as follows. Although the XGBoost approach is known for its higher accuracy performance with better handling capacity of imbalanced data, it is the computationally intensive model when exposed to a larger dataset. It works well for small datasets, but its accuracy starts declining when exposed to streams of data.

A similar problem is also encountered for the SVM approach, and its performance is potentially dependent on an appropriate kernel function selection. Although this selection is somewhat easier for smaller and static sets of data its accuracy starts degrading when assessed with increasing random sizes of incoming traffic of data. The performance of RF is found to be quite good in accuracy for the second dataset, while the DT approach is found to be better for both datasets. However, the DT approach cannot be considered suitable in this implementation scenario as it can lead to a higher degree of instability in case of smaller changes in data. However, the proposed scheme offers better accuracy mainly because of the second algorithm towards the updating process, minimizing all the extra time required to find optimal results. Owing to fitment testing with processed features with normal distribution, the detection outcome is quite reliable with respect to its overall accuracy score. The adoption of correlational score between trained representation offers a better form of data representation considering optimal cost function associated with the constructor and reconstructor operator. This is further justified by evaluating its computational efficiency with respect to algorithm processing time results exhibited in Fig. 6.

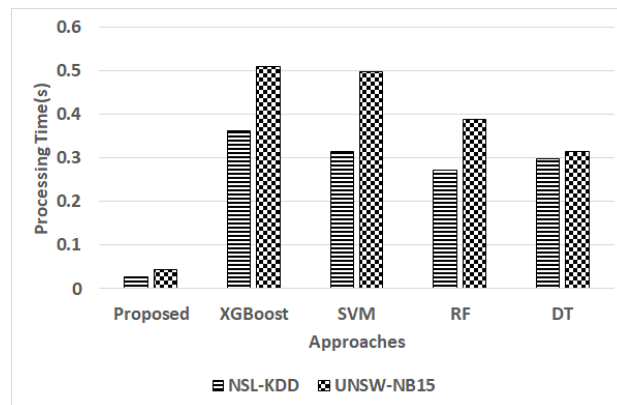


Fig. 6. Processing Time Analysis

3.2 Discussion of Algorithm Processing Time (Fig. 6)

The quantified outcome shows that the proposed system offers approximately 28% and 38% of reduced algorithm processing time in comparison to the mean of existing machine learning algorithms for the first and second datasets, respectively. The justification behind this outcome is as follows: Both XGBoost and SVM are noted with considerably higher algorithm processing time, which is mainly due to their extensive hyperparameter tuning and slower computational process respectively. For both datasets, their performance is nearly the same. RF algorithm generates a large number of trees that eventually leads to increased complexity especially during the training operation. DT has better performance in contrast to XGBoost, SVM, and RF, which is mainly due to its independence from normalization or feature scaling step; however, when exposed to unindexed data, it is noted to be

prone to overfitting, leading to inferior generalization of unseen data.

However, the methodology involved in the proposed scheme calls for unsupervised learning, while the identification of outliers is easily done by estimating reconstruction error without much dependency on labelled data. Further, the model is highly adaptive to complex patterns of new types of intrusion by finetuning the data, as noted in the second algorithm of the proposed system. Hence, without the inclusion of any sophisticated iterative steps, the algorithm can carry out progressive operations to identify even smaller changes in traffic with higher accuracy. This ground of fact is attributed towards reduced algorithm processing time for the proposed machine learning-based security scheme.

4. Conclusion

The adoption of machine learning algorithms has been frequently witnessed in existing literature towards threat detection on various forms of network systems. Keeping aside the potential capability of machine learning algorithms towards addressing security threats, one of the challenging problems identified is that – a trained machine learning model is not sufficient to mitigate dynamic forms of threats, which is a major gap in the literature. This gap is addressed in proposed system by following notable contribution as follows. (i) proposed scheme implements a highly adaptable and flexible architecture which is capable of identifying dynamic threats with faster response and higher accuracy, (ii) neural network-based learning model has been designed with an inclusion of two additional layers (construction operator and reconstructor operator) which assists towards better form of learning representation of variable and uncertain traffic pattern, (iii) proposed scheme is completely free from any form of human intervention which can undertake its own decision threat detection and updating process, (iv) an innovative concept of masked index has been introduced to address the task of labelling so that an involuntary threat detection scheme can be defined on dynamic environment, and (v) proposed system offers approximately mean of 11% increased accuracy and 33% of reduced processing time when compared with frequently used machine learning towards cyber security.

For real-world applicability, the model needs to be synchronized with existing edge devices for particular service providers of IoT-cloud systems. Various challenges that could surface are handling and processing massive incoming traffic from IoT devices with respect to designated edge devices. The complete system operation is carried out considering fault-tolerant processing carried out within edge devices and servers that execute algorithms in a cloud environment. This dependency could be one of the prime limitations although it will not affect algorithmic performance. Hence. Future work will be continued towards improving the same model with more parametric inclusion of cloud attributes and edge device attributes apart from IoT devices, which was the prime focus of this study. The future work will have an inclusion towards novel formulation of threat prevention strategy considering large IoT environment.

References

- [1]. A.S. Mashaleh, N.F. Ibrahim, M.Alauthman, J Al-karaki, A. Almomani, A. Atalla, A. Gawanmeh, "Evaluation of machine learning and deep learning methods for early detection of internet of things botnets," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 14, no. 4, p. 4732, 2024, doi: 10.11591/ijece.v14i4.pp4732-4744.
- [2]. T. Nagaraj and R.K. Channarayappa, "An efficient security framework for intrusion detection and prevention in internet-of-things using machine learning technique," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 14, no. 2, p. 2313, 2024, doi: 10.11591/ijece.v14i2.pp2313-2321.
- [3]. L. Zhang and L. Wang, "A hybrid encryption approach for efficient and secure data transmission in IoT devices," *J. Eng. Appl. Sci.*, vol. 71, no. 1, 2024, doi: 10.1186/s44147-024-00459-x.
- [4]. T. Rajmohan, P.H. Nguyen, and N. Ferry, "A decade of research on patterns and architectures for IoT security," *Journal of Cybersecurity*, vol. 5, no. 1, 2022, doi: 10.1186/s42400-021-00104-7.
- [5]. K. Kandasamy, S. Srinivas, K. Achuthan, and V.P. Rangan, "IoT cyber risk: a holistic analysis of cyber risk assessment frameworks, risk vectors, and risk ranking process," *EURASIP Journal of Info. Secur.*, vol. 2020, no. 1, 2020, doi: 10.1186/s13635-

020-00111-0.

- [6]. T. Mazhar, D.B. Talpur, T.Al Shloul, Y.Y. Ghadi, I.Haq, I. Ullah, K. Ouahada, H.Hamamm, "Analysis of IoT security challenges and its solutions using artificial intelligence," *Journal of Brain Sci.*, vol. 13, no. 4, p. 683, 2023, doi: 10.3390/brainsci13040683.
- [7]. S. Selvarajan, G. Srivastava, A.O. Khadidos, A.O. Khadidos, M.Baza A.Alsheer, J.C-W Lin, "An artificial intelligence lightweight blockchain security model for security and privacy in IIoT systems," *Journal of Cloud Comput. Adv. Syst. Appl.*, vol. 12, no. 1, 2023, doi: 10.1186/s13677-023-00412-y.
- [8]. M. A. Ferrag, L. Maglaras, and M. Benbouzid, "Blockchain and artificial intelligence as enablers of cyber security in the era of IoT and IIoT applications," *Journal of Sens. Actuator Netw.*, vol. 12, no. 3, p. 40, 2023, doi: 10.3390/jsan12030040.
- [9]. K. Chakraborty, D. Kapila, S. Kumar, Bhupati, N. Shaik, and A. Singh, "Intelligent machine learning based internet of things (IoT) resource allocation," *Int. Conf. RAISE-2023*, Basel Switzerland: MDPI, 2023, p. 73.
- [10]. T. Lai, F. Farid, A. Bello, and F. Sabrina, "Ensemble learning based anomaly detection for IoT cybersecurity via Bayesian hyperparameters sensitivity analysis," *Journal of Cybersecurity*, vol. 7, no. 1, 2024, doi: 10.1186/s42400-024-0
- [11]. C. Liu *et al.*, "Dissecting zero trust: research landscape and its implementation in IoT," *Journal of Cybersecurity*, vol. 7, no. 1, 2024, doi: 10.1186/s42400-024-00212-0.
- [12]. S. Mishra, A. Albarakati, and S.K. Sharma, "Cyber threat intelligence for IoT using machine learning," *Journal of Processes (Basel)*, vol. 10, no. 12, p. 2673, 2022, doi: 10.3390/pr10122673
- [13]. M.E.E. Alahi, A. Sukkuea, F.W. Tina, A. Nag, W. Kurdthongmee, K. Suwannarat, S.C. Mukhopadhyay, "Integration of IoT-enabled technologies and Artificial Intelligence (AI) for smart city scenario: Recent advancements and future trends," *Journal of Sensors (Basel)*, vol. 23, no. 11, p. 5206, 2023, doi: 10.3390/s23115206.
- [14]. T. Narasimhamurthy and G. Hosahalli Swamy, "Insights of machine learning-based threat identification schemes in advanced network system," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 14, no. 4, p. 4664, 2024, doi: 10.11591/ijece.v14i4.pp4664-4674.
- [15]. D. Alsalman, "A comparative study of anomaly detection techniques for IoT security using adaptive machine learning for IoT threats," *IEEE Access*, vol. 12, pp. 14719–14730, 2024, doi: 10.1109/access.2024.3359033.
- [16]. M.A. Elsadig, "Detection of denial-of-service attack in wireless sensor networks: A lightweight machine learning approach," *IEEE Access*, vol. 11, pp. 83537–83552, 2023, doi: 10.1109/access.2023.3303113.
- [17]. Z.E. Ekolle, H. Ochiai, and R. Kohno, "Collabo: A collaborative machine learning model and its application to the security of heterogeneous medical data in an IoT network," *IEEE Access*, vol. 11, pp. 142663–142675, 2023, doi: 10.1109/access.2023.3341837.
- [18]. M. Alqahtani, H. Mathkour, and M.M. Ben Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," *Journal of Sensors (Basel)*, vol. 20, no. 21, p. 6336, 2020, doi: 10.3390/s20216336.
- [19]. H. El-Sofany, S. A. El-Seoud, O.H. Karam, and B. Bouallegue, "Using machine learning algorithms to enhance IoT system security," *Scientific Reports*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-62861-y.
- [20]. M. Salb, L. Jovanovic, N. Bacanin, M.Antonijevic, M.Zivkovic, N.Budimirovic, L.Abualigah, "Enhancing internet of things network security using hybrid CNN and XGBoost model tuned via modified reptile search algorithm," *Journal of Appl. Sci. (Basel)*, vol. 13, no. 23, p. 12687, 2023, doi: 10.3390/app132312687.
- [21]. J.A. Faysal, S.T. Mostafa, J.S. Tamanna, K.M. Mumenin, M.M. Arifin, M.A. Awal, A.Shome, S.S.Mostafa, "XGB-RF: A hybrid machine learning approach for IoT intrusion detection," *Journal of Telecom*, vol. 3, no. 1, pp. 52–69, 2022, doi: 10.3390/telecom3010003
- [22]. C. Ioannou and V. Vassiliou, "Network attack classification in IoT using support vector machines," *Journal of Sens. Actuator Netw.*, vol. 10, no. 3, p. 58, 2021, doi: 10.3390/jsan10030058.
- [23]. K.M. Abuali, L. Nissirat, and A. Al-Samawi, "Advancing network security with AI: SVM-based deep learning for intrusion detection," *Journal of Sensors (Basel)*, vol. 23, no. 21, p. 8959, 2023, doi: 10.3390/s23218959.
- [24]. S. Dalal, U.K. Lilhore, N.Faujdar, S.Simaiya, M.Ayadi, N.A.Almujally, A.Ksibi, "Next-generation cyber attack prediction for IoT systems: leveraging multi-class SVM and optimized CHAID decision tree," *Journal of Cloud Comput. Adv. Syst. Appl.*, vol. 12, no. 1, 2023, doi: 10.1186/s13677-023-00517-4.
- [25]. U. Akram, W.Sharif, M. Shahroz, M.F.Mushtaq, D.G.Arday, E.B.Thompson, I.T. Diez, S.Djuraev, I.Ashraf, "IoTTPS: Ensemble RKSVM model-based Internet of Things threat protection system," *Journal of Sensors (Basel)*, vol. 23, no. 14, p. 6379, 2023, doi: 10.3390/s23146379
- [26]. S. Ben Atitallah, M. Driss, and I. Almomani, "A novel detection and multi-classification approach for IoT-malware using random forest voting of finetuning convolutional neural networks," *Journal of Sensors (Basel)*, vol. 22, no. 11, p. 4302, 2022, doi: 10.3390/s22114302
- [27]. A. Hamarsheh, "An adaptive security framework for Internet of Things networks leveraging SDN and machine learning," *Journal of Appl. Sci. (Basel)*, vol. 14, no. 11, p. 4530, 2024, doi: 10.3390/app14114530.
- [28]. A. Alrefaei and M. Ilyas, "Using machine learning multiclass classification technique to detect IoT attacks in real time," *Journal of Sensors (Basel)*, vol. 24, no. 14, p. 4516, 2024, doi: 10.3390/s24144516
- [29]. J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *Journal of Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00892-y.
- [30]. H.A. Gouda, M.A. Ahmed, and M.I. Roushdy, "Optimizing anomaly-based attack detection using classification machine learning," *Neural Comput. Appl.*, vol. 36, no. 6, pp. 3239–3257, 2024, doi: 10.1007/s00521-023-09309-y.
- [31]. J. Jang, Y. An, D. Kim, and D. Choi, "Feature importance-based backdoor attack in NSL-KDD," *Journal of Electronics (Basel)*, vol. 12, no. 24, p. 4953, 2023, doi: 10.3390/electronics12244953.
- [32]. S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced intrusion detection systems performance with UNSW-NB15 data analysis," *Journal of Algorithms*, vol. 17, no. 2, p. 64, 2024, doi: 10.3390/a17020064.