IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Explainable Artificial Intelligence For Cyber Security

Dr. Achyutha Prasad N
HOD , Professor, Dept of CSE.
East West Institute of Technology
Bengaluru, India

Savitha B
Dept of CSE
East West Institute of Technology
Bengaluru, India

***Abstract*:** As a foundational technology in the security net environment, detection of network intrusions has a great deal of attention and use. Network Intrusion Detection (NID) still has difficulties when it comes to installing on devices with limited resources, despite the tremendous efforts of research & advanced technologies. We provide a lightweight detection of intrusions technique based on knowledge of extraction is called Lightweight Neural Network (LNet), which strikes the balance between efficiency and accuracy by concurrently lowering computing costs and model storage. To be more precise, we stack DeepMax blocks to create the LNet after carefully designing the DeepMax blocks to extract compressed representation effectively. Additionally, in order to compensate for the lightweight network's performance decrease, we apply batchwise knowledge of oneself distillation to regularise training consistency. Our suggested Lightweight Neural Network (LNN) and XGboost methodology is shown to be successful on Allflowmeter_Hikari datasets through experiments.

I. INTRODUCTION

How far malicious detection techniques have advanced in the realm of Intrusion Detection Dystems (IDS) space is difficult to gauge. Machine learning-based intrusion detection systems (IDSs) must be trained using the available datasets, however it might be challenging to find a trustworthy dataset to compare. A few of the things that hinder the process of comparing datasets include inadequate method documentation [1], a lack of a comparison method [2], and the absence of critical aspects like ground-truth labelling, publicly accessible traffic, and real-world environment data. Furthermore, only a small number of datasets accurately reflect the fact that network traffic is primarily encrypted these days to preserve security and privacy. The dataset is a crucial component in the development of IDS models based on machine learning. The first step in the procedure is to gather internet traffic, either in the form of packets or flows. Subsequently, the recorded traffic is assembled into a particular kind of data that has attributes associated to networks, such as labelling. Figure 1 depicts a generic machine learning-based IDS. An essential step for the dataset is labelling. Managing ground truth is extremely difficult, particularly when specialists are unable to identify when the traffic is benign or malicious. Researchers employ synthetic traffic for this reason. This suggests, nonetheless, that the created traffic is not typical of the actual world. To put it briefly, gathering traffic is the first step in creating a dataset, and the last step is preprocessing. A labelled dataset is the end product of the preprocessing stage. Every data point is categorised as benign or malevolent. The file provides tabular data in binary form (IDX file) or human-readable format (CSV file). The dataset may be benchmarked based on the quantity of malicious activity or false alarms discovered. The current datasets are not realistic enough to serve as the foundation for developing a complete model for the identification of novel attacks, nor do they contain consistently encrypted traces. The majority of the research that has been done so far using encrypted traffic is concentrated on various domains, such traffic analysis and categorization [3]. Despite the existence of this research [4], this dataset is not accessible to the general public because of data sensitivity.

Benchmark datasets serve as a crucial foundation for assessing and contrasting the quality of various IDS. There are three different types of intrusion detection systems (IDS) based on the methodologies used for detection: signature-based, anomaly-based, and hybrid. The KDD99 dataset is no longer in use, and all three of these kinds of IDS use it to assess their systems. While the anomaly-based approach concentrates on identifying an outlier from the authentic profile, the signature-based approach focuses on developing automatic signature creation [5–6]. The signature-based type recognises and tries to match against the signatures database using a pattern-matching technique. An alert is triggered when the signature of an attempted attack is matched. The most accurate and least likely to cause false alarms is the signature-based kind; nevertheless, it is not able to identify unknown threats. The ratio of alarms that are false is still large, even though the anomaly-based type may be able to identify unknown assaults by comparing anomalous traffic with regular traffic. In this study, we describe a tool and specifications for creating a new dataset in a real-world setting through the generation of encrypted network traffic. We are contributing in two ways. Firstly, we provide new specifications for building new

datasets. Secondly, we build a novel intrusion detection system dataset that includes encrypted traces of network activity. The dataset has attacks labelled, including probing and brute force login. The ground-truth data, background traffic, and packet traces with message are all supplied.The following portions of this essay are arranged as follows: We provide relevant literature in Section II; we build the system description and outline the optimisation issue in Section III. The two-phase alternate optimisation strategy is developed in Section IV, and its efficacy is assessed in Section V. This attempt is eventually concluded in Section VI.

The contributions of this survey are as follows: – We provide a comprehensive background with the main concepts, existing methods, limitations, and risks associated with securing explainable systems. – We discuss open research problems and identified multiple research avenues for future work.

II. Literature Survey

IoT device security is highly vulnerable because of the rise in cyberattacks. With the combined use of machine learning algorithms for the identification and detection of various assaults, the state of the art offers several options for their prevention. Some of the work in this direction is covered in this section.

Title	Authors	Year	Objectives	Advantages	Disadvantages
"A survey on explainable artificial intelligence , (XAI): Toward medical XAI," IEEE transactions on neural networks and learning systems.	E. Tjoa and C. Guan	2020	Transaction on neural networks and learning systems.	XAI provides transparency into AI models, helping users understand how decisions are made. This transparency can enhance trust in AI systems, particularly in critical domains like medicine where trust is paramount.	Impleme nting XAI techniques can be complex and resource-intensive, requiring expertise in both AI and domain-specific knowledge. Developing interpretable models or generating explanations may add computational overhead and increase development time and costs.
"Xai for cybersecurity: state of the art, challenges, open issues and future directions,"	G. Srivastava, R. H. Jhaveri, S. Bhattacharya, S. Pandya, P. K. R. Maddikunta, G. Yenduri, J. G. Hall, M. Alazab, T. R. Gadekallu et al.,	2022	XAI provides insights into the decision-making process of AI-based cybersecurity systems.	enabling cybersecurity professionals to better understand how threats are detected and mitigated.	Impleme nting XAI techniques in cybersecurity systems can be complex and resource-intensive, requiring expertise in both AI and cybersecurity. Developing interpretable models or generating explanations may add computational overhead and increase development time and costs.
"Explainable artificial intelligence in cybersecurity"	N. Capuano, G. Fenza, V. Loia, and C. Stanzione,	2022	XAI can help reduce false positives by explaining why certain events or activities were flagged as potential threats, allowing cybersecurity analysts to distinguish between genuine security incidents and benign events.	XAI facilitates compliance with regulatory requirements such as GDPR and CCPA by providing explanations for AI-driven cybersecurity decisions, ensuring transparency and accountability in data processing..	XAI techniques may inadvertently amplify biases present in the data or the model itself, leading to potentially biased interpretations of cybersecurity events or decisions. Addressing biases in XAI models is crucial to ensure fair and reliable cybersecurity analysis
"Explainable artificial intelligence for cybersecurity"	F. Charmet, H. C. Tanuwidjaja, S. Ayoubi, P.-F. Gimenez, Y. Han, H. Jmila, G. Blanc, T. Takahashi, and Z. Zhang,	2022	The paper provides a literature survey on explainable artificial intelligence (XAI) specifically in the context of cybersecurity, offering a thorough examination of existing research in this domain.	- It likely includes insightful analysis and discussions on the advantages, limitations, and future directions of XAI in cybersecurity, helping readers gain a deeper understanding of the current state of the field.	The paper may lack original research contributions and primarily focuses on synthesizing existing literature, potentially offering limited novel insights or findings

III. System Model

We go into great depth on the suggested lightweight of network intrusion detection in this section. The novel and automated Deep Learning-based attack detection system described here, shown in Figure 1, learns from data collected by the host Internet of Things (IoT) network and, once sufficiently trained, identifies network intrusions. The suggested Intrusion Detection System, or IDS, dynamic connector connects the simulated network to requests coming from the Internet of Things (IoT) network. Through an interface module, the feature extraction and network classifier are in communication with the simulated network. The network packets that make up the dataset underlying the neural network that is used in the suggested technique are extracted of their features by the feature extractor. Through the classifier's Updated module, the suggested IDS is constantly and continuously updated according to newly found characteristics. The network classifier forwards the intrusion to the mitigation step upon detection. The effect of the incursion is lessened at this time.

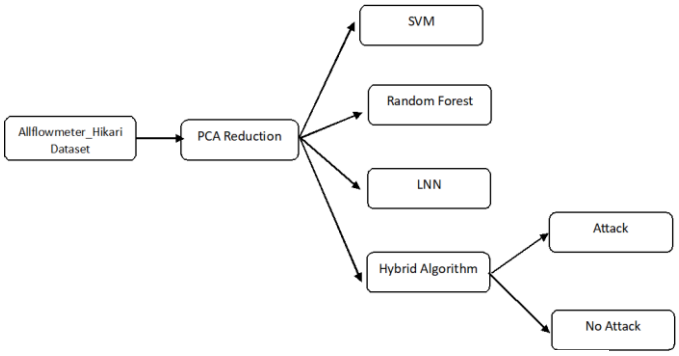

Fig1: The overview of the proposed methodology.

A. Dataset:

One factor that makes evaluating malware detection systems challenging is the dearth of current datasets that are made accessible to the public. The Allflowmeter_HIKARI-2021 dataset, which includes benign traffic and encrypted simulated assaults, is presented in this publication. The content criteria, which centre on the final dataset, & the process necessities, which centre on the dataset's construction, are both met by this dataset. To facilitate the generation of new datasets, we have compiled these prerequisites.

B. Data pre-processing

Six categories—flow features, time characteristics, content amenities, connection features, multipurpose features, and labelled features—are created from the total of 42 labelled features. This study takes into account seven other kinds of assaults in addition to standard data: analysis, fuzzing, code for shells, worms, denial-of-service, exploits, and backdoor assaults. This study employs 5,55,278 records from the dataset for the purpose of training set plus 4,44,222 records as the testing set. Pre-processing of the data can reduce the amount of raw information to expedite the training of the model. The main factors that influence the quality of data are accuracy, consistency, and integrity. But inaccurate, lacking, and inconsistent data can be found in databases and data warehouses in the actual world. The suggested study pre-processes the unstructured information in order to convert it into an organised form after data collecting. Data is divided into tests and training sets as part of the pre-processing stage. Twenty percent of the data is used for testing and eighty percent is used for training in the suggested study. This is the point at which the data begins to exhibit instances of duplication and overlap. A circumstance known as "data duplication" happens when an info sequence appears more than once in a collection. Conversely, data overlapping refers to the situation where a data sequence occurs in both sets. Overlapping and duplicate data might lead to an untrustworthy assessment model. The performance of the model as a whole may be jeopardised if there are sequences in the data pool that overlap. This might happen if the same sequence appears in both the sets used for training and testing. The suggested approach makes advantage of data cleaning to lessen this problem by making sure that there are no duplicate or overlaid data sequences. The original, uncleaned data is kept apart from the clean training and pristine testing data sets.

C. Feature extraction

The process of removing significant characteristics from the dataset is known as feature extraction. This is a crucial step since it minimises redundant data features, saves storage space, and speeds up calculations. Choosing a suitable 0/1 string—where 1 denotes the acceptance of a certain feature and 0 denotes its rejection—is the first step in the feature selection process. The amount of features in the dataset is equal to the length of the string.PCA Dimension Reduction: After all processed records are entered, PCA will choose all pertinent features and eliminate all superfluous features in order to reduce dimensionality. Reduced variables will be divided into groups to train and to test, with 80% of the dataset used for training while 20% for algorithm testing in the application.

D. Training and testing Model

In this study, a lightweight neural network termed an LNN is introduced. It uses a compressed neural network from CONV1D, which required less processing power and resources than CONV2D and 3D. The author uses the dimensionality reduction method of PCA (principal component analysis) to deal with large numbers of parameters by selecting only the most significant characteristics and ignoring the rest.The proposed LNN method has been compared by the author with SVM, for example Random Forest, the proposed CNN, and CNN with or without PCA. Since it is challenging to put into practice all the algorithms, we are using SVM, Random Forest, and proposing LNN with Xgboost here.

IV. Algorithm

In this section, we provide a two-stage alternating optimisation approach that is multitask-based and capable of effectively solving the two a fore mentioned subproblems.

A. Linear SVM

In 1970, SVM was created using ideas from the theory of statistical learning [1]. In essence, it addresses regression and two-class problem with classification. A hyper-plane establishes a categorization border between two classes. Support vectors are the closest points to the hyperplane, and the support vector algorithm (SVM) is the method used to calculate them.

Hyperplane is stated as follows in Eqn 1:

$$w \cdot y + b = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

where w and b stand for the input vector's weight and bias, respectively, and y is the input vector.

SVM is represented mathematically as Eqn 2 :

$$\text{If } w \cdot y + b \geqslant 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(2)$$

then

$$h(x_i) = +1; \text{ otherwise, } h(xi) = -1 \ldots\ldots\ldots\ldots\ldots\ldots(3)$$

Categories A and B are denoted by +1 and -1 in this instance. The following is the final decision Eqn (3)

When data can be split using just one line and are preferred for a high number of features, the linear kernel form of SVM is employed. The necessary kernel formula [1] can be added to the final decision equation.

B. Random Forest

Random Forest is a powerful ensemble learning algorithm widely used in intrusion detection systems due to its ability to handle high-dimensional data, handle missing values, and provide robust performance. In the context of intrusion detection, Random Forest constructs multiple decision trees during training, where each tree is built using a subset of the training data and a random subset of features.One of the key advantages of Random Forest is its ability to handle complex and non-linear relationships within the data. Each decision tree in the forest independently learns to classify instances based on a random subset of features, and the final prediction is determined by aggregating the predictions of all trees. This ensemble approach helps mitigate overfitting and improve the generalization ability of the model.Moreover, Random Forest provides built-in mechanisms to assess feature importance, which can be valuable for identifying the most relevant features for intrusion detection. By analyzing feature

importance scores, analysts can gain insights into the underlying patterns of network traffic data and prioritize features for further investigation.

Random Forest also offers robustness to noise and outliers in the data, making it suitable for handling real-world network traffic with varying levels of complexity and noise. Additionally, the algorithm is computationally efficient and scalable, making it feasible for deployment in real-time intrusion detection systems deployed in Internet of Things (IoT) environments.

Furthermore, Random Forest can handle imbalanced datasets commonly encountered in intrusion detection tasks, where the number of normal instances far exceeds the number of intrusion instances. The algorithm's inherent ability to balance class distributions and handle skewed data helps improve the detection of rare intrusion events while maintaining a low false positive rate.

C. Lightweight Neural Network

Lightweight neural networks for intrusion detection represent a specialized application of machine learning techniques to enhance cybersecurity in computer networks. Unlike traditional intrusion detection systems that may rely on rule-based approaches or heavy computational models, lightweight neural networks are designed to efficiently process network traffic data while minimizing computational resources and memory usage.

These networks are typically characterized by simplified architectures optimized for streamlined feature extraction from network packets. The architecture often includes layers tailored to the characteristics of network traffic data. For instance, convolutional layers may be utilized to capture spatial patterns in packet headers or payload data, while recurrent layers can model temporal dependencies within traffic flows.One key aspect of lightweight neural networks for intrusion detection is their efficiency in training and inference. Training algorithms are designed to optimize the network parameters efficiently, minimizing computational overhead and training time. During inference, the network can quickly process incoming network traffic data in real-time, enabling timely detection of potential intrusions.

Moreover, lightweight neural networks exhibit adaptability to dynamic network environments and evolving threat landscapes. They can dynamically adjust their parameters based on changes in network conditions and adapt to emerging intrusion techniques. Techniques such as online learning or transfer learning may be employed to facilitate continuous model updates and knowledge transfer from related tasks or domains.Robustness to noise and adversarial attacks is another essential characteristic of lightweight neural networks for intrusion detection. These networks are designed to handle noisy data and remain resilient in the face of deliberate attempts to evade detection. Techniques such as data augmentation, regularization, and adversarial training may be employed to enhance the network's robustness and generalization performance.Furthermore, lightweight neural networks are scalable and deployable across various network environments, including edge devices, cloud servers, and network appliances. Efficient deployment mechanisms and model compression techniques enable their deployment in distributed network infrastructures without significantly impacting system performance.

D. XGBoost

XGBoost, short for Extreme Gradient Boosting, is a powerful and popular machine learning algorithm known for its efficiency, scalability, and high performance in various domains, including intrusion detection in computer networks. XGBoost belongs to the ensemble learning family and is particularly well-suited for classification tasks, making it a valuable tool for detecting and classifying network intrusions.

At its core, XGBoost combines the strengths of gradient boosting algorithms with a scalable and optimized implementation, allowing it to handle large datasets with high dimensionality effectively. The algorithm works by iteratively building an ensemble of decision trees, where each subsequent tree is trained to correct the errors of the previous ones. This iterative process continues until a predefined number of trees (or until convergence) is reached, resulting in a highly accurate and robust predictive model.

One of the key advantages of XGBoost is its ability to handle diverse types of data and feature representations commonly encountered in network intrusion detection tasks. Whether the features are categorical, numerical, or a mix of both, XGBoost can effectively learn from them and capture complex patterns in the data. Additionally, XGBoost automatically handles missing values, reducing the need for extensive data preprocessing.XGBoost offers several hyperparameters that can be fine-tuned to optimize model performance, including tree depth, learning rate, regularization parameters, and the number of trees in the ensemble. Through careful hyperparameter tuning, XGBoost can achieve exceptional performance on intrusion detection tasks, balancing between model complexity and generalization ability.Another advantage of XGBoost is its computational efficiency, which is crucial for real-time intrusion detection systems deployed in network environments. XGBoost's parallelized and optimized implementation enables fast training and inference, making it suitable for processing large volumes of network traffic data in real-time.

Moreover, XGBoost provides interpretable results, allowing analysts to understand the underlying patterns and decision-making process of the model. Feature importance scores generated by XGBoost can help identify the most influential features for intrusion detection, aiding in the interpretation and explanation of detected anomalies.

V. Experiment Results

The complete dataset was transformed to numeric representation on the screen, and the final two lines show that, prior to PCA, the dataset included 555278 records and 85 features or columns. Now, select "PCA Dimension

Reduction" to minimise features and obtain the output shown below.

After using PCA, we obtained 20 features out of 43 on the screen above. We can also see the total number of records used for testing and training. Click the "Run SVM Algorithm" button to activate the SVM and obtain the output below.

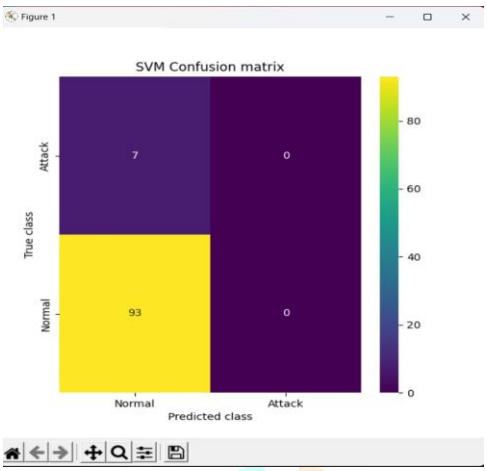

Fig 2: SVM Algorithm Confusion Matrix

The above fig 2 show confusion matrix for Support Vector Machine algorithm which has the accuracy of 95 %

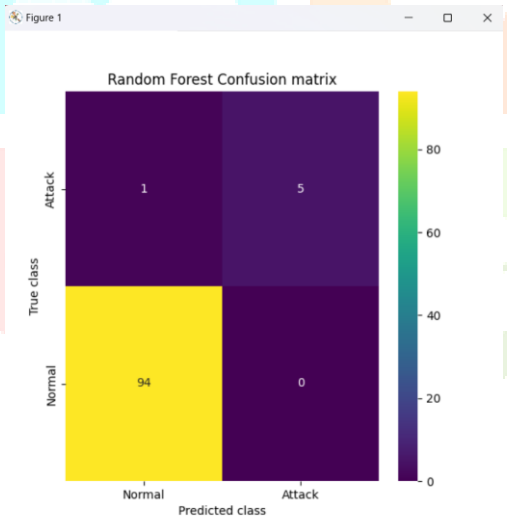

Fig 3: Random Forest Algorithm Confusion Matrix

The above fig 2 show confusion matrix for Random Forest algorithm which has the accuracy of 100 %

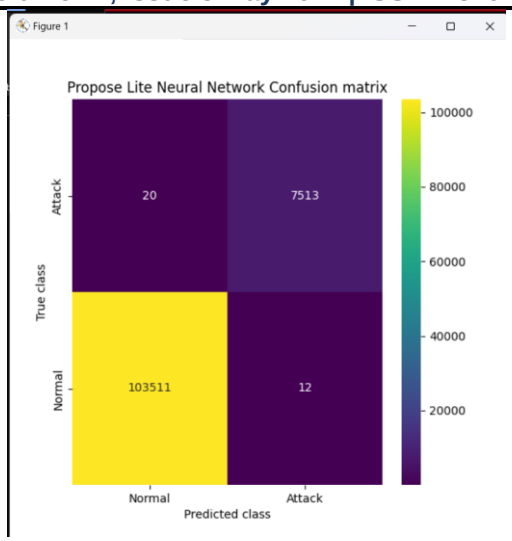

Fig 4: LNN Algorithm Confusion Matrix

The above fig 4 show confusion matrix for Lightweight Neural Network algorithm which has the accuracy of 100.96 %

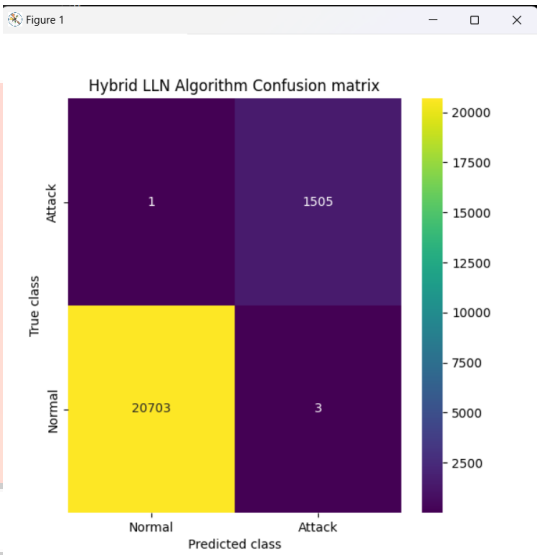

Fig 5: Hybrid Algorithm Confusion Matrix

The above fig 5 show confusion matrix for Hypbrid algorithm which has the accuracy of 100.98 %

Table 1: Comparison of Algorithms

Algorithm	Accuracy	Precision	Recall	F1-Score
SVM	95	48	51	49.4
Random Forest	100	100.4	92.66	96.1
LNN	100.97	100.91	100.86	100.8
Hybrid	100.98	100.89	100.95	100.92

In the above table 1 we can find LNN and Hybrid algorithm gives the accuracy of 100 percent. So we can consider this methods to predict Intrusion under Internet of Things IOT model

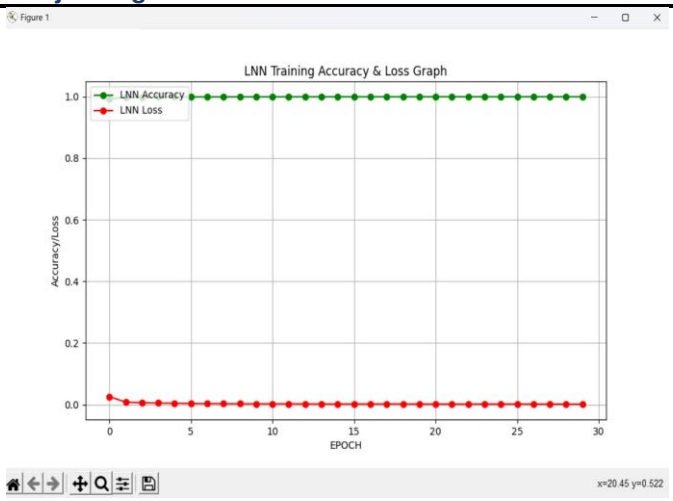

Fig 6: LNN Accuracy and Loss Graph

On the graph, the x-axis denotes the training epoch, the y-axis shows accuracy and loss, and the red and green lines, respectively, show accuracy and loss. As the period progressed, accuracy increased and approached 1, while loss decreased and approached 0.

VI. Conclusions

This work presented LNN, a lightweight detection of intrusions method, for edge devices with low resources. Our suggested technique achieves a superior trade-off between effectiveness and precision when compared to a series of circuits that employed the same topology as LNN. LNet-SKD specifically achieves a near 100% reduction in computational cost overall parameter size while maintaining a little higher accuracy & F1 score. Additionally, LNN performs better than the standard models and other current attack detection theories, which is quite possibly the best outcome at such a cheap resource cost. We find that our suggested method achieves a notable edge during network intrusion detection. We highlighted several research avenues to improve the security of explainable methods, covering both practical aspects such as privacy concerns and ethical aspects, including fairness and fair washing. We conclude this survey by reframing that AI will be a major actor in enforcing business policies and assisting with important decision-making matters. As such, XAI should guarantee fair, clear, and unbiased treatment.

References

1. Jan, S.U.; Ahmed, S.; Shakhov, V.; Koo, I.: Toward a lightweight intrusion detection system for the internet of things. IEEE Access 7, 42 (2019)
2. Nivaashini, M.; Thangaraj, P.: A framework of novel feature set extraction based intrusion detection system for internet of things using hybrid machine learning algorithms. In: 2018 International conference on computing, power and communication technologies (GUCON). pp. 44–49 (2018)
3. Tait, K.-A.; Khan, J. S.; Alqahtani, F.; Shah, A. A.; Khan, F. A.; Rehman, M. U.; Boulila, W.; Ahmad, J.: Intrusion detection using machine learning techniques: an experimental comparison. In: IEEE International congress of advanced technology and engineering (ICOTEN)
4. Khan, M.A.; Khan, M.A.; Latif, S.; Shah, A.A.; Rehman, M.U.; Boulila, W.; Driss, M.; Ahmad, J.: Voting classifier-based intrusion detection for IOT networks. In: 2nd International conference of advanced computing and informatics (ICACIN) (2021)
5. Abiodun, O.I.; Abiodun, E.O.; Alawida, M.; Alkhawaldeh, R.S.; Arshad, H.: A review on the security of the internet of things: challenges and solutions. Wireless Pers. Commun. (2021). https://doi.org/10.1007/s11277-021-08348-9.
6. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2, 12 (2019)
7. Kumari, A.; Mehta, A.: A hybrid intrusion detection system based on decision tree and support vector machine. In: 2020 IEEE 5th International conference on computing communication and automation (ICCCA), pp. 396–400, (2020)
8. Pokharel, P.; Pokhrel, R.; Sigdel, S.: Intrusion detection system based on hybrid classifier and user profile enhancement techniques. Int. Workshop Big Data Inf. Secur. 2020, 137–144 (2020)
9. Kilincer, I.F.; Ertam, F.; Sengur, A.: Machine learning methods for cyber security intrusion detection: datasets and comparative study. Comput. Netw. 188, 107840 (2021)
10. Fitni, Q.R.S.; Ramli, K.: Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. In: 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and communications Technology (IAICT), pp. 118–124. (2020)
11. Fitni, Q.R.S.; Ramli, K.: Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. In 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and communications Technology (IAICT), pp. 118–124. (2020)
12. Krishnaveni, S.; Vigneshwar, P.; Kishore, S.; Jothi, B.; Sivamohan, S.: Anomaly-based intrusion detection system using support vector machine. In: Dash, S.S., Lakshmi, C., Das, S., Panigrahi, B.K. (eds.) Artificial Intelligence and Evolutionary Computations in Engineering Systems, pp. 723–731. Springer Singapore, Singapore (2020)
13. Liang, C.; Shanmugam, B.; Azam, S.; Jonkman, M.; Boer, F.; Narayansamy, G.: Intrusion detection system for internet of things based on a machine learning approach (2019)
14. Yang, L.; Cai, M.; Duan, Y.; Yang, X.: Intrusion detection based on approximate information entropy for random forest classification. In: Proceedings of the 2019 4th international conference on big data and computing. ser. ICBDC 2019. New York, NY, USA: Association for Computing Machinery, p. 125-129 (2019)
15. Kachavimath, A.V.; Nazare, S.V.; Akki, S.S.: Distributed denial of service attack detection using naïve bayes and k-nearest neighbor for network forensics, In 2020 2nd International conference on innovative mechanisms for industry applications (ICIMIA), pp. 711–717, (2020)
16. Verma, A.; Ranga, V.: Machine learning based intrusion detection systems for IOT applications. Wireless Pers. Commun. 111(4), 2287–2310 (2020)
17. Hindy, H.; Bayne, E.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Bellekens, X.: Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset)' (2020)
18. Sah, G.; Banerjee, S.: Feature reduction and classifications techniques for intrusion detection system. Int. Conf. Commun. Signal Process. 2020, 1543–1547 (2020)
19. Latah, M.; Toker, L.: An efficient flow-based multi-level hybrid intrusion detection system for software-defined networks. CCF Trans. Netw. 3(3), 261–271 (2020)
20. Abdulrahman, A.; Ibrahem, M.K.: Evaluation of ddos attacks detection in a new intrusion dataset based on classification algorithms. Iraqi J. Inf. Commun. Technol. 1, 49–55 (2019)
21. Pangsuban, P.; Wannapiroon, P.: A real-time risk assessment for information system with cicids2017 dataset using machine learning. Int. J. Machine Learn. Comput. 10(3), 465–470 (2020)