Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 231 (2024) 40–47

Procedia Computer Science

www.elsevier.com/locate/procedia

The 14th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2023)
November 7-9, 2023, Almaty, Kazakhstan

Cloud Computing Security and Deep Learning: An ANN approach

Lumbardha Hasimi[a,*], Dimitrios Zavantis[b], Elhadi Shakshuki[c], Ansar Yasar[b]

[a] *Comenius University in Bratislava, Slovakia*
[b] *Transportation Research Institute (IMOB), Hasselt University, Diepenbeek, Belgium*
[c] *Jodrey School of Computer Science, Acadia University, Nova Scotia, Canada*

Abstract

Deep learning techniques have shown significant impact in enhancing security across various domains by leveraging artificial neural networks models. When applied to cloud computing security, deep learning offers cost-effective solutions by automating threat detection, reducing manual monitoring, and improving overall security effectiveness. Deep learning models using neural networks play crucial role in tasks like intrusion detection, malware detection, anomaly detection, and log analysis. Integration of deep learning into cloud security requires careful evaluation of existing systems, defining objectives, dataset selection and preparation, model tuning, and eventual modifications for compatibility. Furthermore, implementing deep learning techniques in cloud security entails considering factors such as computational resources, data collection and preparation costs, model development, integration efforts, and ongoing monitoring and maintenance. This paper proposes a feed-forward propagation Artificial Neural Network (ANN) model in cloud security and investigates the key steps for integrating such models into cloud security strategies. Considering that the effectiveness of the ANN model depends on factors such as training data quality, network architecture, and weight adjustment algorithms, the study utilizes a dataset from Kaggle.com for validation and demonstrates steps involved in training and evaluation of the ANN model.

Keywords: cloud security, deep learning, ANN, network

* Corresponding author.
E-mail address: hasimi2@uniba.sk

10.1016/j.procs.2023.12.155

1. Introduction

Deep learning (DL) techniques have emerged to become effective tools for enhancing security in various domains. These techniques leverage the capabilities of artificial neural networks to learn and identify patterns from vast amounts of data, enabling more robust and efficient security solutions [1][2]. When applied to cloud computing security, deep learning techniques offer several benefits having direct implications in the cost-effectiveness of the products. A great benefit of implementation of DL in security of cloud is seen in automated threat detection. DL algorithms can analyze large volumes of data, such as network traffic logs, system logs, and user behavior, to detect anomalies and potential security threats automatically [2], [3]. This together with other utilizations as a result of DL, reduces the need for manual monitoring and analysis, what leads to faster identification of security incidents, timely response, mitigation, and reduced costs in time and resources. Considering this, through automation of various security tasks, DL minimizes the risk of human error, which can lead to security breaches and associated costs.

Automated systems powered by deep learning can perform tasks consistently and accurately, enhancing overall security effectiveness [4], [5]. DL models learn patterns and relationships in data, allowing for more accurate threat detection and classification. Through detecting patterns and anomalies which could go unnoticed by traditional rule-based systems, DL implementation makes it possible for organizations to optimize resource allocation and minimize unnecessary costs.

However, with the emerging trends and developments, real-time detection remains the most important and challenging issue for any system. To tackle this problem, it is necessary to be able to offer solutions that can analyze data in real-time and enable prompt identification of potential security threats[6]. Considering that DL models are able to adapt to evolving threats and learn from new data in real-time, such an approach becomes the most effective tool for optimization. By minimizing response times, avoiding false positives, and reducing the need for frequent manual updates and configuration, the security system becomes more efficient and cost-effective. Moreover, analyzing historical data and patterns, DL provides valuable insights into potential future threats. Through predictive analytics, organizations can proactively allocate resources and implement preventive measures, mitigating potential security incidents and their associated costs [7], [8].

Apart from above mentioned benefits, there are other non-technical benefits to be taken into account. Namely, qualitative factors, such as improved reputation, customer trust, competitive advantage etc. These factors should also be part of the big picture as they contribute to the overall value of adopting deep learning techniques [1], [8].

As seen from the stated benefits, deep learning techniques offer cost-effective solutions to cloud computing security by various means. Integrating deep learning can optimize resource allocation, enhance overall security effectiveness, and mitigate potential financial losses.

2. Integrating Deep Learning approach to cloud security

Deep learning models and algorithms play a crucial role in enhancing cloud computing security. These models are utilized in various cloud security applications, including intrusion detection, malware detection, anomaly detection, log analysis, access control etc. [2], [3], [9]. As already mentioned, the biggest value of DL models lies in the ability to learn complex patterns, detect anomalies, and adapt to evolving threats. However, when it comes to choosing the specific models for the cloud security, it fully depends on the nature of the problem, availability of computational resources, data, sensitivity and the architecture of the existing system and other requirements case-specific to the organization. Some of the commonly used deep learning models and algorithms in the context of cloud security include CNN, RNN, LSTM, GAN, DRL etc.

Organizations can integrate deep learning techniques into their cloud security strategies in several ways. As already discussed, organizations can have multiple benefits from integration of deep learning in cloud computing security [10]. Regardless of the application in tasks such as Data Analysis, Anomaly Detection, Malware Detection and Classification, Intrusion Detection, Prevention, User Authentication, and Access Control, the successful implementation of deep learning techniques requires adequate strategies and resources [2], [11], [12].

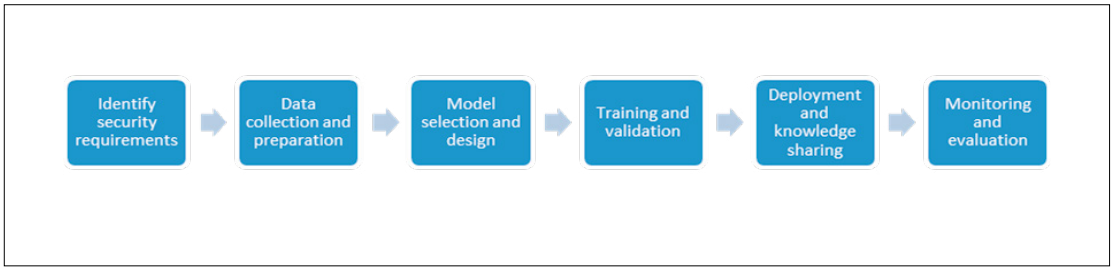

Fig. 1. Key steps to integrate deep learning into cloud security.

In Figure 1 are shown key steps to integrate deep learning into cloud security. By following these steps, organizations can successfully integrate deep learning into their cloud computing security strategies.

However, for the organizations that want to integrate the deep learning in existing solutions and architectures of security, the process would require a more thorough evaluation (Figure 3). The first phase towards implementation requires assessment of strengths and weaknesses of the existing system. This would allow easier identification of areas and tasks that might require or allow DL integration.

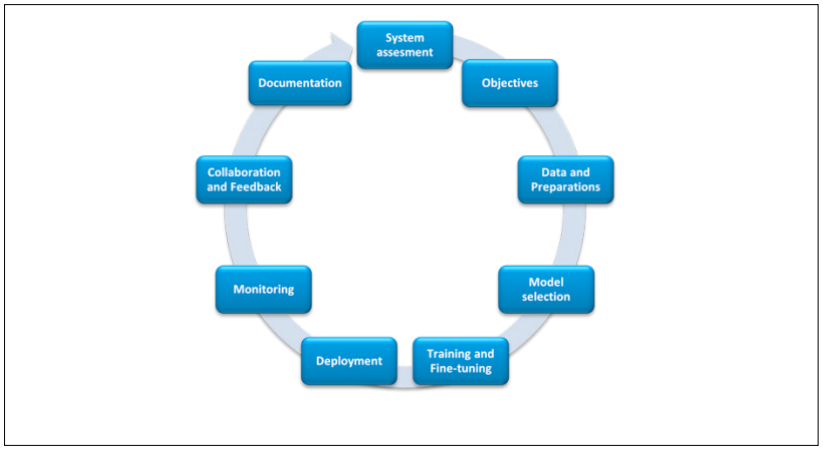

Fig. 2. Key phases for DL integration into existing security systems

The phase of finding the areas and tasks that could be enhanced using DL is closely related to pre-defined objectives or goals the organization has. Having clearly defined objectives, to determine the specific tasks or challenges that deep learning can address within the cloud security system is very important for the optimization of strategies. The latter has direct indication to the model selection that is suitable for the identified tasks and purposes. Training the model requires vast amount of data that are pre-processed and carefully selected for the given task [13]. Given the importance of dataset selection, data processing and preparation, this phase requires a thorough investigation. The whole success of the model depends on dataset and data preparation[14]. After deciding on the data, to adapt to the specific cloud security context it is needed to consider the tuning of the model including adjusting hyperparameters, conducting cross-validation, optimization etc. The final integration process might need eventual modifications to ensure compatibility and integration with existing components. Monitoring and feedback are part of the circle of successful integration, therefore having the right mechanisms planned for this stage is necessary.

3. Neural Network and the proposed model

A neural network at an advanced level can be known as a computational model that over an arrangement of layers depict inputs to outputs with interrelated processing units[13], [15]. Such a model filters the data that passes over the network to determine the relevant input signal that will be passed to the next layer. It basically decides whether or not a particular neuron will be activated, and in the absence of specific neurons, it develops as a plain linear model[16].

Aiming to determine the effect of factors on cloud security costs, we present a new approach using a feed forward propagation ANN model. Based on homogeneous encryption technology, the suggested method may immediately train and create a simple ANN model for encrypted data[13]. For the implementation of the approach there are three phases: the training phase, the testing phase, and the validation phase, which are linked across a cloud environment.

ANNs are trained using labeled data, where the input features are known, and the corresponding outputs are provided for learning[15]. The training process involves presenting the input data to the network, computing the output, comparing it with the expected output, and adjusting the weights and biases through back propagation. The ANN method can be applied to various tasks, including classification, regression, pattern recognition, and time-series forecasting[13]. Its effectiveness depends on factors such as the quality and quantity of the training data, the choice of network architecture, and the optimization algorithm used for weight adjustment.

3.1. Dataset

The dataset used for this study is available at Kaggle.com (The dataset which is used to check the validity of the proposed methodology is available at https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot). The dataset includes 52 files with 1638 columns with various features. The dataset includes features obtained from different sources, including alerts, system resources, logs, network traffic, and proposed new 61 features with high correlations from 1176 features[14]. A dataset of an ANN model consists of two main components the input data and the output labels.

The dataset is usually divided into three subsets for training, validation, and testing. The training subset is used to train the ANN model. It comprises a large portion of the dataset and is used to optimize the network's weights and biases through the learning process. The training graph for this has been produced using MATLAB shown in Figure 3. The validation subset is used to tune the hyper parameters of the ANN model during the training process. It helps to assess the model's performance on unseen data and make decisions on adjustments. The validation graph has been produced using MATLAB shown in Figure 4b. The test subset is used to evaluate the final performance of the trained ANN model. It provides an unbiased assessment of the model's generalization capabilities on unseen data[15]. The testing graph has been produced using MATLAB as shown in Figure 4a.

3.2. Pre-processing

Preprocessing is essential preparing the dataset for training an ANN model. It involves transforming and manipulating the raw input data to make it suitable for the network to learn from[17]. Preprocessing helps improve the quality of the data, normalize the features, and handle any inconsistencies or missing values. Some of the most common pre-processing techniques include data cleaning, data normalization, and data augmentation. These preprocessing techniques help in improving the quality and suitability of the dataset for training an ANN model, ensure fair representation of all features, handle missing values, and address data-related challenges specific to the problem domain[18].

In our dataset, prior to primary processing, the sample data and group data are formulated to build the set-up of the ANN model. After dataset selection, three basic issues are dealt with. The missing data is the first issue, and this data is interchanged by the ordinary immediate value, then normalized and randomized the data. A mean method used to calculate the missing values is formulated as:

$$T(c) = \begin{cases} mean\ (c),\ if\ \ c = null \\ c,\ otherwise \end{cases} \tag{1}$$

3.3. Application layer in ANN

In ANN model, the application layer refers to the last layer of neurons, where each neuron corresponds to a specific output class or value [19]. The application layer's purpose is to transform the information learned by the previous layers into a form suitable for the problem. It provides the final predictions or outputs of the ANN model. In this ongoing analysis, there are 19 neurons in the input layer, whereas the output layer contains 16 hidden neurons. The ANN model is designed to signify only one objective defining the effect of different parameters on cloud cost security.

Sigmoid (x) function of input layer is defined by (2) capable of inscribing the Activation (x) function, where the input hidden layer is defined by (3).

$$S(x) = \frac{1}{1 + e^{-\partial j}} \tag{2}$$

$$\partial j = \sum_{i=1}^{m} \left(\alpha_i j + E_i \right) + b_1 \tag{3}$$

Similarly, the Activation (x) function of output layer is defined by (4) and the respective output hidden layer is defined by (5).

$$A(x) = \frac{1}{1 + e^{-\partial k}} \tag{4}$$

$$\partial k = \sum_{j=1}^{n} (\beta_j k + E_i) + b_2 \tag{5}$$

In the above equations, b_1 and b_2 are the arbitrary constant, and E_i represents the errors associated with the problem. Whereas αij and βij define the direct associative degree of the input layer and the output layer respectively, while ∂j and ∂k are the measured values of the input and output hidden layers.

3.4. Validation layer in ANN

During the training phase of an ANN, a subset of the available labeled data is left for validation purposes. This subset, known as the validation set, is not used for training the network but rather serves as an independent dataset to evaluate the model's performance during the training process[20]. The validation set is used to monitor the model's generalization and to tune hyper parameters or make decisions on model optimization[21]. During the training process, the model was periodically evaluated on the validation set to measure its performance using precision and regression. Shown in Figure 4, the validation result of the dataset using the neural network. We applied this to validate the performance of the network. By training and testing datasets concerning targets, the regression plots show the outputs of the network.

4. Simulation and Results

The datasets were subjected to a machine learning technique, and simulations were carried out using the MATLAB tool. The dataset is split 70% used for training, while the remaining 30% for testing and validation. The data has been encrypted and stored in the cloud. To assess the performance of the proposed model, we trained it on different epochs and used the MATLAB program to apply three distinct ANN algorithms: Levenberg-Marquardt (LM), Bayesian Regularization (BR), and Scale Conjugate Gradient (SCG). The simulation findings in Figure 3, and Figure 4 illustrate the trained dataset model on the cloud using ANN, a state regression plot and training evolution and the outcomes achieved.

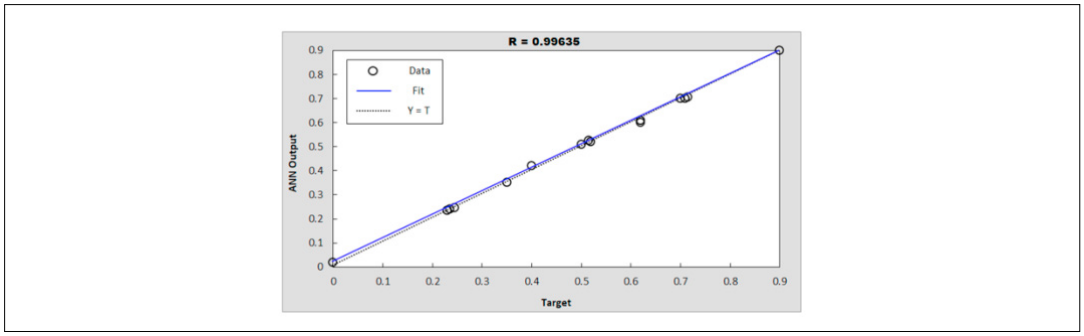

Fig. 3. Regression plot for training

Figure 4 depicts the cloud validation/test result of the dataset using a NN technique. This was used to validate the network performance. The regression graphs demonstrate the network's outputs after training and testing using target datasets. The data fall along a 45-degree line, a perfect fit, with the targets equal to the network outputs. The fit is practically good for all data sets, with R values higher than 0.95; hence possible to retrain the datasets for more accurate findings. Considering the initial load and preferences, the network's output will alter after retraining, affecting the results to further improvement.

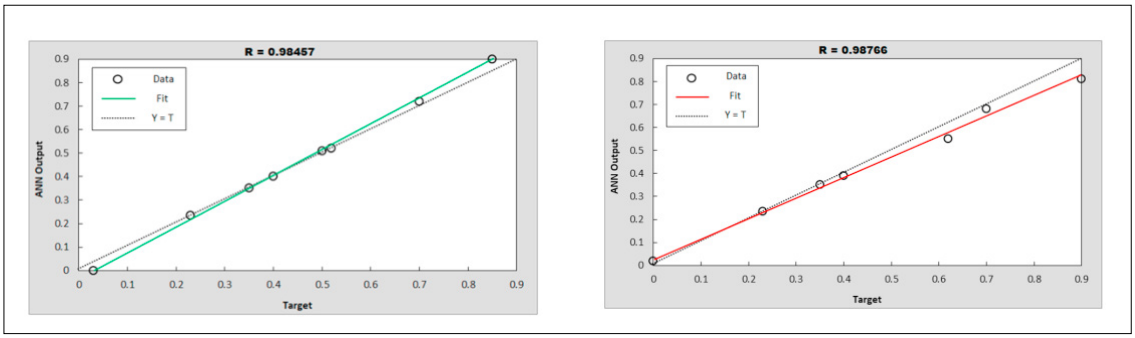

Fig. 4. (a) Regression for testing; (b) regression for validation.

5. Conclusion

The integration of deep learning techniques into cloud computing security offers numerous benefits. By analyzing large volumes of data, deep learning algorithms can detect security threats in real-time and minimize various risks. However, the implementation of deep learning in cloud security requires careful evaluation of costs and resources. Organizations struggle when having to adjust the solution to the computational resources, data acquisition and preparation costs, personnel expertise, and ongoing monitoring and maintenance.

The integration of deep learning techniques in cloud security involves few steps for a successful outcome. These include assessing the strengths and weaknesses of the existing system, identifying specific tasks that deep learning can address, selecting suitable models and algorithms, training the models with labeled data, and ensuring compatibility and integration with existing components. Continuous monitoring and feedback are also crucial stages for maintaining the efficiency and effectiveness of the integrated system.

To give an overview of the simple task implementing a deep learning technique, we proposed a feed-forward propagation Artificial Neural Network (ANN) model in cloud security. The model was trained and developed for encrypted data, enabling efficient analysis for the evaluation of the efficiency in cloud security. The effectiveness of the model depends on factors such as the quality and quantity of training data, network architecture design, and optimization algorithms. The integration of deep learning techniques and the implementation of ANN models offer cost-effective solutions for enhancing cloud computing security, improving threat detection, and optimizing resource allocation. A continuation of this work will further focus on certain effective elements in the overall costs of the security in cloud, being analyzed separately and as parts of a larger group classification. Future research should explore the development of efficient deep learning architectures specifically optimized for cloud computing security. This implies designing architectures that can handle large-scale data processing, reduce, and minimize computational and resource requirements. Addressing such objectives and research directions, deep learning techniques can continue to advance and provide cost-effective solutions and enhance security in cloud computing.

References

[1] E. K. Subramanian and L. Tamilselvan, ‘A focus on future cloud: machine learning-based cloud security’, *SOCA*, vol. 13, no. 3, pp. 237–249, Sep. 2019, doi: 10.1007/s11761-019-00270-0.

[2] N. Srikanth and T. Prem Jacob, ‘An Real Time Cloud Security System and Issues comparison using Machine and Deep Learning’, in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India: IEEE, Nov. 2021, pp. 523–529. doi: 10.1109/I-SMAC52330.2021.9640650.

[3] S. Badri *et al.*, ‘An Efficient and Secure Model Using Adaptive Optimal Deep Learning for Task Scheduling in Cloud Computing’, *Electronics*, vol. 12, no. 6, p. 1441, Mar. 2023, doi: 10.3390/electronics12061441.

[4] K. Gulen, ‘Artificial Intelligence And Automation: Examples, Benefits And More’, Dec. 09, 2022. https://dataconomy.com/2022/12/09/artificial-intelligence-and-automation/ (accessed May 30, 2023).

[5] K. W. Ullah, A. S. Ahmed, and J. Ylitalo, ‘Towards Building an Automated Security Compliance Tool for the Cloud’, in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, Melbourne, Australia: IEEE, Jul. 2013, pp. 1587–1593. doi: 10.1109/TrustCom.2013.195.

[6] H. Xu, ‘Cybersecurity and Data Quality in Cloud Computing: A Research Framework’, in *Information Systems*, M. Papadaki, P. Rupino da Cunha, M. Themistocleous, and K. Christodoulou, Eds., in Lecture Notes in Business Information Processing. Cham: Springer Nature Switzerland, 2023, pp. 201–208. doi: 10.1007/978-3-031-30694-5_15.

[7] S. Moisset, ‘How Security Analysts Can Use AI in Cybersecurity’, *freeCodeCamp.org*, May 24, 2023. https://www.freecodecamp.org/news/how-to-use-artificial-intelligence-in-cybersecurity/ (accessed Jun. 27, 2023).

[8] A. B. Nassif, M. A. Talib, Q. Nasir, H. Albadani, and F. M. Dakalbab, ‘Machine Learning for Cloud Security: A Systematic Review’, *IEEE Access*, vol. 9, pp. 20717–20735, 2021, doi: 10.1109/ACCESS.2021.3054129.

[9] D. Chauhan, A. Kumar, P. Bedi, V. A. Athavale, D. Veeraiah, and B. R. Pratap, ‘An effective face recognition system based on Cloud based IoT with a deep learning model’, *MICROPROCESSORS AND MICROSYSTEMS*, vol. 81, Mar. 2021, doi: 10.1016/j.micpro.2020.103726.

[10] N. Kryvinska and L. Bickel, 'Scenario-Based Analysis of IT Enterprises Servitization as a Part of Digital Transformation of Modern Economy', *Applied Sciences*, vol. 10, no. 3, Art. no. 3, Jan. 2020, doi: 10.3390/app10031076.
[11] R. Zarai, M. Kachout, M. A. G. Hazber, and M. A. Mahdi, 'Recurrent Neural Networks and Deep Neural Networks Based on Intrusion Detection System', *OALib*, vol. 07, no. 03, pp. 1–11, 2020, doi: 10.4236/oalib.1106151.
[12] M. Kawai, K. Ota, and M. Dong, 'Improved MalGAN: Avoiding Malware Detector by Leaning Cleanware Features', in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Okinawa, Japan: IEEE, Feb. 2019, pp. 040–045. doi: 10.1109/ICAIIC.2019.8669079.
[13] M. U. Sana, Z. Li, F. Javaid, H. B. Liaqat, and M. U. Ali, 'Enhanced Security in Cloud Computing Using Neural Network and Encryption', *IEEE Access*, vol. 9, pp. 145785–145799, 2021, doi: 10.1109/ACCESS.2021.3122938.
[14] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, 'Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning'. TechRxiv, Jan. 27, 2022. doi: 10.36227/techrxiv.18857336.v1.
[15] M. M. Taye, 'Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions', *Computation*, vol. 11, no. 3, Art. no. 3, Mar. 2023, doi: 10.3390/computation11030052.
[16] A. Gupta and M. Kalra, 'Intrusion Detection and Prevention system using Cuckoo search algorithm with ANN in Cloud Computing', in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Nov. 2020, pp. 66–72. doi: 10.1109/PDGC50313.2020.9315771.
[17] W. Etaiwi and G. Naymat, 'The Impact of applying Different Preprocessing Steps on Review Spam Detection', presented at the 8TH INTERNATIONAL CONFERENCE ON EMERGING UBIQUITOUS SYSTEMS AND PERVASIVE NETWORKS (EUSPN 2017) / 7TH INTERNATIONAL CONFERENCE ON CURRENT AND FUTURE TRENDS OF INFORMATION AND COMMUNICATION TECHNOLOGIES IN HEALTHCARE (ICTH-2017) / AFFILIATED WORKSHOPS, E. Shakshuki, Ed., 2017, pp. 273–279. doi: 10.1016/j.procs.2017.08.368.
[18] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, 'A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data', *Frontiers in Energy Research*, vol. 9, 2021, Accessed: Jun. 27, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenrg.2021.652801
[19] P. E. Rauber, S. G. Fadel, A. X. Falcão, and A. C. Telea, 'Visualizing the Hidden Activity of Artificial Neural Networks', *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 101–110, Jan. 2017, doi: 10.1109/TVCG.2016.2598838.
[20] G. B. Humphrey *et al.*, 'Improved validation framework and R-package for artificial neural network models', *Environmental Modelling & Software*, vol. 92, pp. 82–106, Jun. 2017, doi: 10.1016/j.envsoft.2017.01.023.
[21] R. Pramoditha, 'Why Do We Need a Validation Set in Addition to Training and Test Sets?', *Medium*, Apr. 12, 2022. https://towardsdatascience.com/why-do-we-need-a-validation-set-in-addition-to-training-and-test-sets-5cf4a65550e0 (accessed Jun. 27, 2023).