European Journal of Computer Science and Information Technology, 12 (9), 10-22, 2024
Print ISSN: 2054-0957 (Print),
Online ISSN: 2054-0965 (Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

Adaptive Neuro-Fuzzy Model for Enhanced Keylogging Attack Mitigation

Mariam Gbadegesin[1], Solomon Akinola[2], Bukola Owolabi[3]

[1]Department of Computer Science, Faculty of Natural and Applied Sciences,
Lead City University, Ibadan, Oyo State, Nigeria

[2]Department of Computer Science, Faculty of Natural and Applied Sciences,
University of Ibadan, Ibadan, Oyo State, Nigeria

[3]Department of Computer Science, Faculty of Natural and Applied Sciences,
Lead City University, Ibadan, Oyo State, Nigeria

doi: https://doi.org/10.37745/ejcsit.2013/vol12n91022 Published December 02, 2024

Citation: Gbadegesin M., Akinola S., Owolabi B. (2024) Adaptive Neuro-Fuzzy Model for Enhanced Keylogging Attack Mitigation, *European Journal of Computer Science and Information Technology*, 12 (9), 10-22

Abstract: *Keylogging malware is a grave risk to user credentials and data integrity in the constantly evolving discipline of cybersecurity. In an effort to conquer this obstacle, our study optimizes early keylogging detection by creating a Neuro-fuzzy prediction model based on keystroke dynamics. The model was trained on a dataset of more than 500,000 keystroke samples from real keyloggers and simulated users by combining adaptive neural networks and fuzzy logic inference. The tailored Neuro-fuzzy model clearly reduced false positives, increasing accuracy to 99.62% and precision to 66.67, compared to the initial neural networks' 99.1% detection accuracy. A 0.378 MSE is a performance indicator that highlights the model's resilience. By identifying unusual keystroke patterns prior to leaking data, our technology offers an early defense against keylogging, which is a major improvement over conventional defensive defense.*

Keywords: neuro-fuzzy, neuro-fuzzy model, keylogging, keylogging threats

INTRODUCTION

In the digital age, with the widespread use of computers and the internet, the threat of cyber-attacks has grown exponentially. One of the prominent threats that continue to pose serious risks to personal and organizational data is keylogging attacks. Keylogging attacks involve malicious software or hardware that silently records a user's keystrokes, capturing sensitive information, such as passwords, credit card numbers, and personal messages. Researchers have put in diverse efforts in order to reduce or mitigate the malicious activities. Researchers have work on mitigating keylogging attack through analysis of user behavior to detect keylogging using behavioral biometrics approach, encryption of keystrokes at the keyboard driver level, and application of several machine learning technique to analyze user behavior in order to detect keylogging. All these are done in order to stay ahead of keylogging attack and to protect sensitive information of users.

While secure trying to protect against keylogging attacks, there are various challenge encountered by the existing solutions. Challenges such as employment of elusion techniques by sophisticated keyloggers, impact on user experience through network latency [1]. Also, compatibility with various network [2] is also one of the major drawbacks. This study ensures early and reliable detection of keylogging attacks by utilizing adaptive neural networks and fuzzy logic inference.

A keylogger, whether in the form of hardware or software, is a malicious program that can record every keystroke made on a compromised device. By logging these keystrokes, a keylogger can capture sensitive and private information, presenting a severe cyber security threat. It grants unauthorized access to cybercriminals who can exploit the data for malicious purposes such as identity theft, financial fraud, or other harmful activities. With the aid of a keylogger, an attacker can automatically record keyboard inputs, allowing them to access private information stored in secure databases without needing to physically break into a location [1].

Traditional methods of detecting and mitigating keylogging attacks have often fallen short due to the rapidly evolving sophistication of these attacks. Conventional anti-virus and intrusion detection systems may not effectively recognize new and emerging keyloggers, making it essential to develop more intelligent and adaptive approaches to protect against such threats. Euro-fuzzy systems, a combination of artificial neural networks and fuzzy logic, have emerged as a promising technique in the field of cyber security. This hybrid approach leverages the strengths of both neural networks and fuzzy logic to build robust and adaptive predictive models. Neural networks excel in pattern recognition and handling complex, non-linear relationships, while fuzzy logic is excellent at dealing with uncertainty and imprecision in data.

The Neuro-Fuzzy Predictive Model for Keylogging Attack Mitigation is a proactive approach aimed at mitigating the risk of keylogging attacks by predicting and preventing potential threats before they can cause harm. The model was trained on historical data of known keylogging attacks, legitimate user behavior, and system logs to learn the patterns and characteristics of both malicious and benign activities. The Neuro-Fuzzy Predictive Model provides a proactive and adaptive approach to keylogging attack mitigation, offering an extra layer of security to protect sensitive data from potential threats. By continuously learning and adapting from new data, the model can stay ahead of emerging keylogging techniques, ensuring a higher level of protection in today's ever-evolving cyber-threat landscape.

The task of identifying advanced keyloggers has become increasingly challenging for anti-virus software and anti-malware solutions. Unlike traditional viruses and worms, these enhanced keyloggers are highly elusive and hard to detect, posing a significant challenge in terms of detection and prevention. The most concerning aspect of keyloggers arises when they are the result of third-party interference. In such cases, these malicious entities breach computer systems and surreptitiously obtain various types of information, which they subsequently share with other parties for illegal purposes [2].The rise in keylogger attacks can be attributed to several factors, including the limited research focused on simulating patterns that match keylogger signatures. Insufficient investigation in this area makes it difficult to identify and counter new variants effectively. Moreover, the lack of a truly effective detection method for keyloggers further exacerbates the problem. As a result, cybercriminals are finding it easier to deploy these stealthy threats, leading to an increasing number of successful attacks [3], [4].

A lot of keyloggers are written in C or C++ and are executable files. To install them on the system, administrator authorization is needed. It may be hard to find them in the task manager. Usually, the log files that they produce on the system are hidden or designed to look like standard operating system files [5].

Keyloggers are primarily classified into two main categories: hardware keyloggers and software keyloggers. Hardware keyloggers are convenient to use since they are placed within the computer's internal hardware or discreetly inserted between the CPU and the keyboard wire. However, to install a hardware keylogger, the cybercriminal needs physical access to the computer system when no one is observing. Numerous methods, including statistical methods, neural networks, genetic algorithms, support vector machines, fuzzy logic, and hybrid approaches combining neural networks with genetic algorithms, neural networks with support vector machines, and neuro-fuzzy networks, can be used to predict the model of software threats.

Neuro-fuzzy computing offers a combined solution, incorporating the system identification and interpretability of fuzzy models along with the learning capabilities of neural networks. Over the past decade, numerous neuro-fuzzy systems have been created. The neuro-fuzzy based approach involves learning rules and membership functions from the provided data. Neuro-fuzzy is categorized as an adaptive network, comprising nodes and directional links, where some or all nodes have adjustable parameters influencing their outputs. One commonly used learning rule for these adaptive networks is backpropagation. This research develops a Neuro-fuzzy predictive model using keystroke dynamics to reliably detect and mitigate ongoing keylogging threats.

RELATED WORKS

Keylogging remains a prevalent cyber threat that enables attackers to silently capture sensitive information by recording users' keystrokes. A keylogger attack is a sort of cyberattack in which keystrokes on a target device are recorded using software. Attacks of this nature can be used to steal private information, such as login information and credit card details. Keylogger attacks are frequently directed against certain people or groups, and the attackers may be aware of the systems and configuration of the target in advance. Depending on the kind of information the attacker is attempting to steal, they will choose one of the several keylogger attack techniques available. An attacker might, for instance, set up a hardware keylogger on the target's computer to log each keystroke [6]. An alternative would be for the attacker to create malicious software that records keystrokes and sends them to a remote server. Keylogger attacks are challenging to spot since the keylogger software can be passed off as a legitimate program or run covertly in the background. There are several indicators, nevertheless, such as odd computer activity or strange network traffic, indicating a keylogger attack may be underway. Using a reliable antivirus application and keeping all software updated are the best ways to guard against keylogger attacks. Users should also exercise caution when opening attachments or clicking on hyperlinks that originate from unidentified sources [6].

Developing effective techniques to detect and prevent keylogging attacks is an active research area. Neuro-fuzzy systems have emerged as a promising approach combining neural networks and fuzzy logic to create intelligent hybrid models for classification and prediction tasks. This literature review analyses prior research efforts on applying neuro-fuzzy modeling specifically for keylogging attack mitigation

A seminal study by Kolter and Maloof [7] first examined using machine learning for keylogging detection. They extracted n-graph stylometric features capturing typing patterns from raw keystroke data. Experiments compared various classifiers including support vector machines and showed stylometric n-graphs could reliably distinguish users and detect mimicked data from a keylogger. This early work demonstrated the feasibility of keystroke dynamics for keylogging detection.

Building on this, Halevi and Saxena [8] explored using neural networks for user authentication and keylogging detection based on typing rhythms. They modeled temporal patterns using duration and latency features and developed a Hopfield neural network classifier. Testing showed significantly higher accuracy than prior work along with the capability to incrementally train the model with new user data. The authors discussed integrating fuzzy logic to further improve learning from sparse training data.

Ahmed and Traore [9] presented one of the first implementations of a neuro-fuzzy system for anomaly detection using keystroke dynamics. A Mamdani fuzzy inference system modeled rules capturing distinct typing patterns. A self-organizing neural network then optimized the membership functions and model parameters. Evaluation using the GREYC keystroke dataset showed 95% accuracy in classifying genuine and imposter test users. The neuro-fuzzy system outperformed other classifiers such as k-nearest neighbours.

Pillai and Siddavatam [10] discussed a technique for detecting keyloggers, which are programs that can monitor all activities carried out on a PC and steal sensitive information. The proposed technique involves using a machine learning algorithm called support vector machine (SVM) to determine the presence of keyloggers. The algorithm separates the keyloggers from other programs by marking them positive if they have predefined functions and negative if they do not. The study also provides references to related research on keyloggers and computer security.In another study, a method for detecting abnormal patterns of network connections is discussed that combines artificial neural networks, immune systems, and neuro-fuzzy classifiers. It is suggested to use principal component analysis to solve the given problem more effectively. Based on the use of the suggested methodologies, the intrusion detection system's architecture is presented. The primary benefit of the created solution to intrusion detection is a multi-level analysis technique: initially, a mixture of adaptive detectors is used, followed by a signature-based analysis. Numerous computational experiments are carried out. These tests show that the chosen techniques are effective in terms of false positive, true positive, and accurate classification rates [11].

In order to distinguish between programs with appropriate access and keylogger applications that may abuse permissions, a study by Alghamdi *et al* [12] aimed to identify each application's set of rights and storage levels. This keylogger detection method is entirely black-box; it is based on behavioral characteristics that are shared by all keyloggers and does not depend on the internal organization of the keylogger. In this study, a model for detecting keyloggers and spyware using machine learning has been proposed. To recognize the host behaviour while a keylogger is operating on the system, the model was trained using spyware and keylogger data sets. To determine the effectiveness of the system in detecting keylogger spyware, the results was assessed using a variety of metrics and reported based on the classification report and confusion matrix.

The research done by Belej and Halkiv [13] focuses on the development of a network attack detection system using hybrid neuro-fuzzy algorithms. It starts by discussing the shortcomings of existing intrusion

detection methods and the importance of identifying new types of network attacks. It introduces the concept of an intrusion detection system (IDS) and its role in monitoring networks for malicious activity. The research emphasizes the changing nature of network attacks and the need for new detection schemes, including hybrid and adaptive approaches.

A study presents CaFISKLD, which automatically simulates keylogger patterns with ASCII-coded sequences and uses a back-to-back combinatorial algorithm for keylogger detection and analysis. The system also uses a fuzzy inference system to categorize keyloggers into their severity levels. The system was evaluated and the authors found that it outperformed other keylogger detection methods in terms of accuracy and efficience [14].

According to Pradeepthi and Kannan [15], intruders are increasingly attacking different networks, and one of the most popular methods for doing so is through using botnets. The detection and elimination of bots from a network has grown to be highly challenging for network administrators. In another research article, a comprehensive fuzzy-based computational method was introduced for selecting an efficient approach to detect malicious network traffic. This research aimed to address the increasing demand for an intelligent and accurate system for detecting malicious traffic, especially in light of emerging cyber threats [16]. The proposed mechanism employed a systematic approach called fuzzy-AHP TOPSIS to assess the impact of various factors during the performance evaluation stage of implementation.

The study involved the participation of 70 security experts from various software companies and academic institutions. They provided their insights on the criteria and linguistic values involved in the evaluation process. The performance of six different approaches for detecting malicious network traffic was evaluated using the integrated fuzzy-AHP-TOPSIS method. The results of the evaluation indicated that MTD4 was the most effective approach, followed by MTD5, MTD6, MTD1, MTD2, and MTD3.The study concludes that the proposed mechanism can help in enhancing cybersecurity by providing an efficient and effective way of detecting and classifying malicious traffic [16].

Due to the increasing prevalence of cybercrimes, it has become crucial to detect and assess security risks associated with acquiring data from emerging technologies. This is essential for gaining an understanding of how these technologies could potentially be exploited or misused [17]. In order to effectively counter online attacks, it is imperative to develop a distinctive strategy for assessing cybersecurity risks. In this research, we propose utilizing the fuzzy inference model (FIS) to generate risk assessment outcomes. This assessment is based on four key risk variables: vulnerability, threat, likelihood, and impact. These variables help define the spectrum of risks that could potentially jeopardize any entity, and the aim is to address and resolve such issues for the entities under consideration. They have conducted numerous evaluations on these issues, and the outcomes of the study ultimately demonstrate the strength of our suggested course of action [17].

As a consequence of the increase in network security vulnerabilities, network security administrators are now more concerned with identifying the possible attack path of an attacker and fixing flaws. We put forth a new method for predicting network attack paths, which relies on a combination of a knowledge graph and an attack graph model. This approach aims to address the shortcomings of existing methods that primarily predict attack paths in perfect conditions while ignoring the crucial roles that network nodes play [18]. In this method, the central component is the knowledge graph. It incorporates the

quantitative indicators from CVSS for individual vulnerabilities and combines them with a network security evaluation approach to calculate potential attack paths [18]. Reviewing these prior works, neuro-fuzzy techniques show strong promise for enhancing keylogging detection compared to other machine learning approaches. The integration of fuzzy logic and neural learning provides capabilities to model complex typing behaviours, adapt to new threat patterns, and provide interpretable outputs. However, there remain significant research gaps, such as evaluating robustness against adversarial evasion attempts. As keylogging attacks grow more sophisticated, developing intelligent neuro-fuzzy predictive models offers a potent defensive technique worthy of continued research.

METHODOLOGY

Building of keylogging Attack Mitigation Model requires scalable dataset that is pattern-driven. The dataset gotten from Cyber security and Infrastructure Security Agency (CIC) was used in building this model. Data pre-processing was carried out on the dataset to make it suitable for building an enhanced model.

Dataset Description

The keylogger dataset which provides labeled dataset was used in building the model. The dataset contains different columns representing features such as system performance, user activity, and probable gauges of keylogging software. The dataset contains 526,617 instances and 86 columns. Table 1 shows the description of the dataset used for this study in developing keylogging mitigation model.

Table 1: Data Description

Number of Instances	Number of Features	Benign Classes	Malicious Classes
526,617	86	309,415	214,202

Feature Extraction

Before creating a deep learning model, features are extracted from the dataset. Following characterization, Figure 1 illustrates the different steps involved in extracting every attribute required to build the keylogging attack mitigation model. The process selects all the features, then create a subset from the features which the best features will be used in building the model. Then finally test the model's performance.

Figure 1: Feature extraction model

Developing Neuro-Fuzzy Model

Anaconda with jupyter notebook was set up in order to develop the neuro fuzzy model. Libraries such as pandas, NumPy, scikit-fuzzy, and Keras were installed. The keylogger dataset in csv format is loaded into python which was splitted into training and testing subsets.

Fuzzy logic rules were integrated with neural network layers to design the model. The study begins with an input layer sized according to the dataset's features. Dense hidden layers were added and sigmoid output layer was used for binary sorting. Fuzzy rules were defined based on keylogging features, such as unusual typing or access patterns, using scikit-fuzzy.

The model was training with some parameters such as binary cross-entropy as the loss-function, epoch, and batch size. Finally, the model was evaluated using Deep Learning evaluation metrics such as accuracy, recall, RSME, and precision to ensure effective keylogging detection.

Model Evaluation Metrics

Accuracy, loss function, precision, recall, root mean square error, and other metrics are used in this work to assess how well a neuro-fuzzy model performs and how effective it is on a dataset.

Accuracy

This is the proportion of all datasets that have accurate predictions. It is defined as the ratio of true positives, true negatives, false positives (FP), and false negatives (FN) to the number of true negatives (TN) and true positives (TP). A dataset that the algorithm successfully classifies as true or false is called a true positive and negative, whereas a dataset that the algorithm incorrectly classifies is called a false positive and negative [3]. One of the crucial criteria needed to obtain an accurate performance evaluation analysis is accuracy [4]. The mathematical equation is given as:

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP} \qquad 1$$

Precision

The quality of the positive prediction produced by the model is called precision, which is sometimes referred to as positive predictive value [5][6]. It is the quantity of classified fault-prone datasets that are in fact fault-prone datasets [4]. This performance statistic is calculated by dividing the total number of TP and FP by the number of TN. The quality of the model on the dataset increases with the prediction. However, models naturally trade off precision and recall; that is, the more precise the model, the poorer the recall, and vice versa [6]. The degree of utility of the search results is known as precision [6]. Precision is represented mathematically as:

$$\text{Precision} = \frac{TN}{TP+FP} \qquad 2$$

Where TN signifies True Negative, TP mean True Positive and FP means False Positive

Loss Function

The loss function is the consequence for a poor prediction. The loss function is used to determine how poorly a model predicted a particular occurrence. If the loss is zero, the model's forecast is flawless; if not, the loss is greater [7].

Recall

Recall is the proportion of a class that is properly identified out of all the examples of that class that are provided. How comprehensive the search results are is known as recall [6]. It is computed mathematically by dividing TP by FN, or TP and FN, which should have been positively predicted [8].

$$\text{Recall} = \frac{TP}{TP+FN} \qquad 3$$

Where TP denotes True Positive and FN denotes False Negative.

Root Mean Square

Root Mean Square is abbreviated as RMSE. When assessing how well a regression line fits the datasets, it is helpful. Another way to think about it is as the residual standard deviation [9]. The distance from the regression line indicates how distant the datasets are from the standard deviation, which represents how widely separated the values are; residuals are simply error of prediction [8].

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{yi} - yi)^2} \qquad 4$$

Where N denotes total number of values, yi symbolizes actual value of the data used and $\widehat{yi}$ means predicted values of the data used.

Keylogging Attack Mitigation Model

This section discusses the processes involved in the development of the enhanced neuro-fuzzy model for mitigating keylogging attacks.

Step 1: **Gathering Information and Extracting Features**
The first step involves collecting information on system activity with an eye toward spotting trends that could point to keylogging. This entails examining system performance indicators, mouse activity, and keystroke frequency. After that, pertinent features that might be signs of keylogging are extracted.

Step 2: Generating Rules and Fuzzy Logic
The collected features are classified into different risk levels, including Very High (VH), High (H), Medium (M), Low (L), and Very Low (VL), using fuzzy logic. A value is assigned to each risk level, with VH = 1 to VL = 0. The likelihood of keylogging activity is ranked and prioritized using these values, giving the model a flexible rule foundation.

Step 3: Design of the Adaptive Neuro-Fuzzy Inference System
An Adaptive Neuro-Fuzzy Inference System (ANFIS) recognizes patterns in keylogging behavior by combining fuzzy inference and neural networks. Based on real-time data, ANFIS layers automatically adapt to new attack patterns and learn to identify keylogging activity.

Step 4: Training and Assessing the Model
Historical keylogging data and tuning parameters like learning rate, batch size, and epoch are used to train the model. The model's efficacy is assessed using metrics including accuracy, recall, and F1-score, which help to improve its ability to discriminate between malicious and benign behavior.

Stage 5: Real-Time Monitoring and Prediction
After training, the model performs real-time monitoring to detect and predict potential keylogging attacks. By leveraging historical attack patterns and adapting dynamically, the model not only identifies but also prevents future keylogging events through predictive insights, enhancing overall system security.

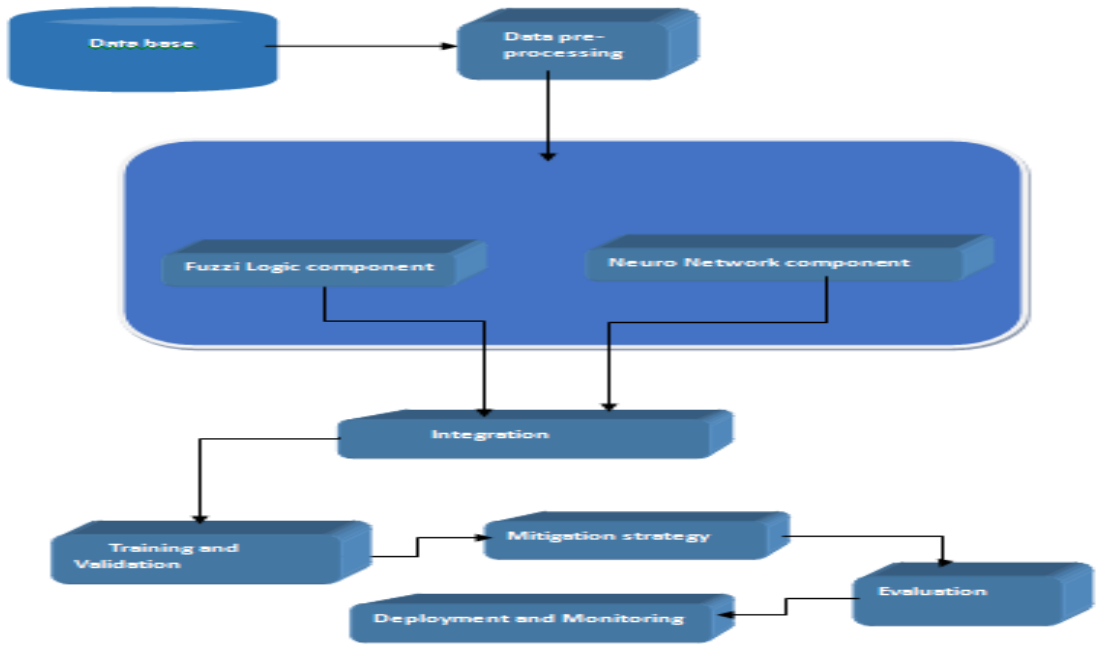

Figure 2: keyloggers prediction Model

Programming language and environment
R is a versatile programming language and software environment primarily used for statistical analysis, creating graphics, and generating reports. It supports integration with procedures written in other languages like C, C++, .NET, Python, or FORTRAN to enhance efficiency. R is entirely free and open-source, with a thriving community of active members.

R includes a package known as the frbs package, which is written entirely in R. This package is designed to implement the most commonly used FRBS (Fuzzy Rule-Based Systems) models, specifically the Mamdani and Takagi Sugeno Kang (TSK) models. Both of these methods are available within the frbs package, allowing users to work with fuzzy logic and fuzzy rule-based systems in R.

R-Studio is an integrated development environment (IDE) designed for the R programming language. It encompasses various features, such as a console, a syntax-highlighting code editor that allows for direct code execution, and a range of tools for tasks like data visualization, version history, debugging, and managing workspaces. R-Studio comes in both open-source and commercial editions and is compatible with desktop operating systems like Windows, Mac, and Linux. It can also run in a web browser when

connected to R-Studio Server.

Algorithms

Neuro-fuzzy systems integrate the adaptive learning capabilities of artificial neural networks with the human-inspired reasoning of fuzzy logic, offering an intelligent hybrid system. In the neuro-fuzzy architecture, layered neural networks that represent membership functions and fuzzy rules [19] model a fuzzy inference system. The core benefit of this approach is enabling fuzzy systems to learn from data, overcoming a weakness of static fuzzy systems dependent on predefined rules [20].

Learning in neuro-fuzzy systems leverages backpropagation algorithms to tune parameters of membership functions to minimize error, fitting inputs to optimal fuzzy sets. The adaptive network-based fuzzy inference system (ANFIS) architecture is a commonly used neuro-fuzzy technique, applying backprop to identify ideal fuzzy if-then rules [21]. This hybrid training approach merges the numerical optimization of neural nets with the linguistic interpretability of fuzzy logic. Neuro-fuzzy principles have been widely applied for classification and prediction problems across domains including time series forecasting, control systems, and pattern recognition.

For cybersecurity applications, neuro-fuzzy systems offer several key advantages. Most importantly, the adaptive learning capabilities allow neuro-fuzzy models to automatically tune their fuzzy rule base and membership functions as new data on threats like keyloggers emerges [22]. This facilitates continuous improvement of detection accuracy. Additionally, the fuzzy rule outputs provide interpretability, unlike the black box models of deep neural networks, enabling understanding of why anomalies are flagged [23] Fuzzy logic is also apt for handling the imprecision of cyber data.

However, challenges include extensive hyperparameter tuning of membership functions, difficult to optimize neural-fuzzy architectures, and lack of certainty in anomalous fuzzy rule firing [24]. Smith *et al.* [25] propose combining neuro-fuzzy systems with evolutionary algorithms to evolve optimal network structure and parameters. Overall, the integration of learning and reasoning makes neuro-fuzzy models a promising AI approach for adaptive security against evolving keylogging threats.

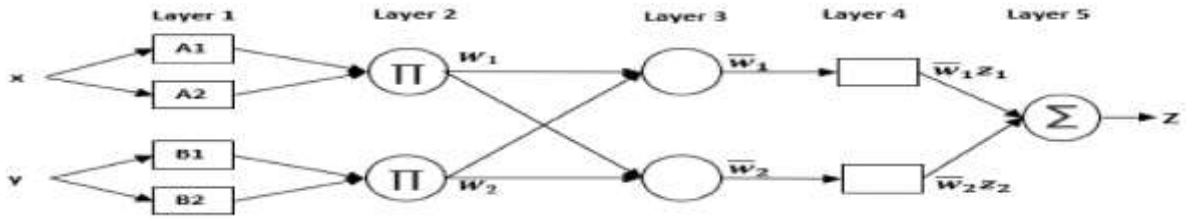

Figure 3: Neuro-fuzzy system

RESULT AND DISCUSSION

This section discusses the performance evaluation metrics for the adaptive neuro-fuzzy model that was employed in this study. Key machine learning benefit metrics, such as accuracy, precision, and recall, were used to evaluate the model's efficacy. These metrics were computed using a 70:30 data splitting ratio and the loss function. Table 2 provides a summary of the findings.

Table 2: Performance Evaluation of Benefit Metrics for Neuro-Fuzzy Model

Splitting Ratio	Accuracy (%)	Precision
Neuro-fuzzy 70:30	99.62	66.67

According to the metrics, the neuro-fuzzy model obtained an accuracy of 99.62%, and flawless precision at 66.67. These outcomes demonstrate how well the model can detect keylogging activity.

Furthermore, for the same data split, cost measure, Mean Squared Error (MSE), were assessed. Table 3 displays the findings, showing the model's overall fit and error rates.

Table 3: Performance Evaluation of Cost Metrics for Neuro-Fuzzy Model

Splitting Ratios	MSE
Neuro-Fuzzy 70:30	0.378

All things considered, the performance metrics demonstrate how well the adaptive neuro-fuzzy model detects keylogging assaults, confirming its potential as a strong instrument in cybersecurity defense tactics.

CONCLUSION

This paper concludes by presenting a thorough method for mitigating keylogging attacks using an adaptive neuro-fuzzy model, proving the value of fusing fuzzy logic with neural networks for cybersecurity applications. Through the use of several phases, including data collection, fuzzy rule generation, adaptive modeling, training, and real-time monitoring, the model offers a strong foundation for accurately detecting, categorizing, and forecasting keylogging actions. By adapting dynamically to new risks, our adaptive neuro-fuzzy model enables proactive mitigation, going beyond conventional detection techniques.

The results highlight the promise of neuro-fuzzy systems in cybersecurity, especially for attack patterns that need a high degree of precision and agility, such as keylogging. Future studies could improve this model's adaptability in a variety of system contexts and broaden its use to include other virus types. All things considered, this study strengthens defenses against changing keylogging attacks by providing a model that can react in real-time to both known and unknown threats.

References

[1] M. Y. Darus, M. Azizi, & M. Ariffin, (2022). Enhancement Keylogger Application for Parental Control and Monitor Children ' s Activities. Journal of Positive School Psychology, 6(3), 8482–8492.

[2] M. Proactive, I. Suitable, C. Enterprises, J. Sun, C. Liu&H. Yuan. (2021). Keylogger Detection and Prevention Journal of Physics: Conference Series, 2007(1). https://doi.org/10.1088/1742-6596/2007/1/012005

[3] F.E. Ayo, S.O. Folorunso, A.A. Abayomi-Alli, A.O. Adekunle, & J.B. Awotunde (2020). Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection. Information Security Journal: A Global Perspective, 29(6), 267-283.

[4] A.A Royo, M.S. Rubio,W. Fuertes, M.C. Cuervo, C.A. Estrada& T. Toulkeridis (2021). Malware Security Evasion Techniques: An Original Keylogger Implementation. In World Conference on Information Systems and Technologies (pp. 375-384). Springer, Cham.

[5] M. Aslam, R.N. Idrees, M.M. Baig, and M.A. Arshad, "Anti-Hook Shield against the Software Key Loggers", National Conference on Emerging Technologies 2004.

[6] Y. Balakrishnan and P. N. Renjith, "An analysis on Keylogger Attack and Detection based on Machine Learning," *2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India*, 2023, pp. 1-8, doi: 10.1109/ICECONF57129.2023.10083937.

[7] 90. J. Z. Kolter and M. A. Maloof, "Learning to detect malicious executables in the wild," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 470-478, 2014. https://www.jmlr.org/papers/volume7/kolter06a/kolter06a.pdf

[8] T. Halevi and N. Saxena, "Detecting mobile malware threats to home computer networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2291-2305, 2013

[9] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," *IEEE Transactions on dependable and secure computing*, vol. 4, no. 3, pp. 165-179, 2014.

[10] D. Pillai and I. Siddavatam, "A modified framework to detect keyloggers using machine learning algorithm," *International Journal of Information Technology*, vol. 11, no. 4, pp. 707–712, 2018. [Online]. Available: https://doi.org/10.1007/s41870-018-0237-6

[11] F. E. AYO, J. AWOTUNDE, S. Folorunso, and O. A. Olalekan, "Cafiskld: A combinatorial-based fuzzy inference system for keylogger detection," *SSRN Electronic Journal*, 2022. doi:10.2139/ssrn.4111802

[12] S. M. Alghamdi, E. S. Othathi, and B. S. Alsulami, "Detect keyloggers by using machine learning," *2022 Fifth National Conference of Saudi Computers Colleges (NCCC)*, 2022. doi:10.1109/nccc57165.2022.10067780

[13] O. Belej and L. Halkiv, "Development of a network attack detection system based on hybrid neuro-fuzzy algorithms," *Computer Modeling and Intelligent Systems*, vol. 2608, pp. 926–938, 2020. doi:10.32782/cmis/2608-69

[14] F. E. AYO, J. AWOTUNDE, S. Folorunso, and O. A. Olalekan, "Cafiskld: A combinatorial-based fuzzy inference system for keylogger detection," *SSRN Electronic Journal*, 2022. doi:10.2139/ssrn.4111802

[15] K. V. Pradeepthi and A. Kannan, "Detection of Botnet traffic by using Neuro-fuzzy based Intrusion Detection," *2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India*, 2018, pp. 118-123, doi: 10.1109/ICoAC44903.2018.8939109

[16] S. H. Almotiri, "Integrated Fuzzy Based Computational Mechanism for the Selection of Effective Malicious Traffic Detection Approach," in *IEEE Access*, vol. 9, pp. 10751-10764, 2021, doi: 10.1109/ACCESS.2021.3050420.

[17] M. Alali, A. Almogren, M. M. Hassan, I. A. L. Rassan, and M. Z. Bhuiyan, "Improving risk assessment model of cyber security using Fuzzy Logic Inference System," *Computers & Security*, vol. 74, pp. 323–339, 2018. doi:https://doi.org/doi:10.1016/j.cose.2017.09.011

[18] Y. Wang, Z. Sun, and Y. Han, "Network Attack Path Prediction Based on Vulnerability Data and Knowledge Graph," *International Journal of Innovative Computing, Information and Control*, vol. 17, no. 5, pp. 1717–1730, Oct. 2021. doi: http://ijicic.org/ijicic-170518.pdf
[19] M. Pratama, R. Hartono, S. Fauziati and J. H. Kim, "Neuro-Fuzzy waterfall model for software development project investment decision," *International Journal of Fuzzy Systems*, vol. 22, no. 1, pp. 13-22, Feb. 2020.
[20] R. Anand and S. Dhande, "Detection and prevention of SQL injection attack using ANFIS," 2019 *3rd International conference on electronics, communication and aerospace technology (ICECA)*, vol. 2, pp. 12-16, 2019.
[21] R. Bello, S. Zeadally, and M. Badra, "Network intrusion detection system for IoT cybersecurity based on learning techniques," IEEE Internet of Things Journal, vol. 7, no. 11, pp. 10280-10289, 2019.
[22] S. J. Han and S. B. Cho, "Evolutionary neural networks for anomaly detection based on the behavior of a program," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 3, pp. 559-570, June 2006.
[23] F. Ahmed, A. Ali, V. S. Sheng, W. Li, A. Aziz, and A. U. R. Khan, "Efficient deep learning model for cyber-attack detection," IEEE Access, vol. 7, pp. 42402-42414, 2019.
[24] D. J. Smith, M. Kitchen, M. Popenici, O. Sudit, R. Torres, D. Tratz-Ryan, M. Venkata, J. Ward, T. Wilson, et al., "Evolving multilayered cyber defenses through an adapted NEAT algorithm," in Military Communications Conference, *MILCOM 2018-2018 IEEE*, pp. 607-612, *IEEE*, 2018.
[25] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in Proceedings, vol. 89. *Presses universitaires de Louvain*, 2019, pp. 89-94.
[1] A. Mohan, A. Singh, B. Kumar, and R. Dwivedi, "Review on remote sensing methods for landslide detection using machine and deep learning," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 7, pp. 1–23, 2021, doi: 10.1002/ett.3998.
[2] J. Kim and H. Lee, "Adaptive human-machine evaluation framework using stochastic gradient descent-based reinforcement learning for dynamic competing network," *Appl. Sci.*, vol. 10, no. 7, pp. 1–15, 2020, doi: 10.3390/app10072558.
[3] S. Rawat, K. P. S. Rana, and V. Kumar, "A novel complex-valued convolutional neural network for medical image denoising," *Biomed. Signal Process. Control*, vol. 69, no. May, p. 102859, 2021, doi: 10.1016/j.bspc.2021.102859.
[4] J. Soni and K. Mathur, "Data mining and machine learning: Design a generalized real time sentiment analysis system on tweeter data using natural language processing," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, pp. 2139–2142, 2019, doi: 10.35940/ijeat.F8492.088619.
[5] S. Achar, "An Overview of Environmental scalability and Security in Hybrid Cloud Infrastructure Designs," vol. 8, no. 2, pp. 39–46, 2021.
[6] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Futur. Gener. Comput. Syst.*, vol. 79, pp. 849–861, 2018, doi: 10.1016/j.future.2017.09.020.
[7] A. Acien, A. Morales, J. V. Monaco, R. Vera-Rodriguez, and J. Fierrez, "TypeNet: Deep Learning Keystroke Biometrics," *IEEE Trans. Biometrics, Behav. Identity Sci.*, vol. 4, no. 1, pp. 57–70, 2022, doi: 10.1109/TBIOM.2021.3112540.
[8] C3 AI, "Root Mean Square Error (RMSE)," *C3 Ai*, 2023. .
[9] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *Int. J. Comput. Appl.*, vol. 44, no. 9, pp. 875–886, 2022, doi: 10.1080/1206212X.2021.1974663.