"JUDGMENT OF AI IMPOSITION UNCOVERING CLASSIFICATION FOR SOFTWARE CLEAR LINKAGE"

[1]Name of 1st Ushaben Barad
[1]Designation of 1st Assistant Professor
[1]Name of Department of 1st Faculty of Engineering
[1]Name of organization of 1st Gokul Global University, Sidhpur, Patan, Gujarat – India

ABSTRACT

This research paper examines the integration of Software-Defined Network (SDN) architecture into Wireless Sensor Networks (WSNs) to enhance security in sensitive environments. Specifically, the focus is on establishing a Software-Defined Wireless Sensor Network (SDWSN). The findings reveal that the decision tree approach outperforms the others and is the most suitable for implementing IDSs in conventional SDWSNs due to its superior performance.

Keywords: AI, intrusion detection, SDWSN, security, WSN.

INTRODUCTION

The integration of AI into Intrusion Detection Systems (IDS) for Software-Defined Networks (SDNs) brings intelligence and adaptability to network security. AI-IDS leverages machine learning algorithms, data analytics, and pattern recognition to proactively detect and mitigate potential security breaches in real-time. Traditional IDS face limitations in the dynamic SDN environment, but AI-IDS can learn from network data, detect new attack patterns, and adjust detection algorithms accordingly. This adaptive nature allows AI-IDS to stay ahead of emerging threats and optimize resource allocation. AI-IDS enhances network performance by reducing false positives and streamlining incident response efforts. However, SDN introduces security challenges, making AI-IDS crucial for safeguarding network infrastructure, protecting data, and ensuring uninterrupted network operation. The key components of SDN include the controller, network devices, southbound and northbound APIs. However, challenges include controller reliability, security concerns, compatibility, skill gap, and the transition from legacy infrastructure. Traditional IDS solutions lack visibility, scalability, and adaptability in SDN. AIbased IDS overcome these limitations by offering enhanced detection accuracy, real-time threat detection, adaptability to changing environments, scalability, and intelligent threat response. The integration of AI with IDS in SDN provides more effective and adaptive security measures. AI-powered IDS leverage machine learning algorithms to analyze data, detect anomalies, and enhance incident response. However, human expertise is still necessary to interpret AI-generated alerts and make informed decisions. Overall, AI has brought significant advancements to IDS, enhancing detection capabilities and improving the security posture in the face of evolving cyber threats.

LITERATURE REVIEW

SDN provides network visibility, flow-based analysis, centralized control, programmability, and integration of AI techniques for IDS. IDS in SDN requires a holistic approach considering these characteristics. Key challenges include visibility and monitoring, dynamic network topology, flow-based analysis, security of the SDN controller, traffic steering, and orchestration. IDS placement can be centralized at the controller, distributed at network devices, or a hybrid approach. Factors to consider for IDS placement include network topology, performance, security policy enforcement, resource utilization, scalability, manageability, and integration with the SDN controller. Flow table analysis is a common technique for intrusion detection in SDN. It involves inspecting flow rules, detecting anomalies, analyzing traffic redirection, detecting policy violations, and performing flow correlation to enhance accuracy. Flow table analysis leverages the visibility and control provided by SDN to detect and respond to intrusions effectively. However, it should be complemented with other intrusion detection techniques like packet inspection.

Intrusion Detection Systems (IDS) in the context of Software-Defined Networking (SDN) rely on traffic monitoring and packet inspection to detect potential security breaches and malicious activities. Key aspects related to traffic monitoring and packet inspection in IDS include:

1. Flow-based Monitoring: SDN organizes network traffic into flows, and IDS analyzes flow-level information to identify deviations from normal traffic patterns and detect potential intrusions. 2. Real-time Traffic Analysis: SDN enables real-time collection and analysis of network traffic data, allowing IDS to detect intrusions immediately and respond in a timely manner. 3. Deep Packet Inspection: IDS performs detailed analysis of packet headers, payload content, and metadata to detect known attack signatures or anomalies, enabling granular network traffic analysis. 4. Traffic Filtering and Sampling: Traffic filtering rules and sampling techniques help manage the volume of network traffic analyzed by IDS, focusing resources on relevant traffic and ensuring efficient utilization. 5. Encrypted Traffic Inspection: IDS employs techniques such as SSL/TLS decryption and traffic decryption proxies to inspect encrypted traffic and detect potential threats within encrypted communication channels. 6. Protocol-specific Inspection: IDS incorporates protocol-specific inspection techniques to identify protocolbased attacks or anomalies by analyzing specific characteristics and vulnerabilities of different network protocols. 7. Anomaly Detection: IDS utilizes anomaly detection techniques, such as machine learning algorithms or statistical models, to establish normal behavior patterns and identify deviations that may indicate intrusions or attacks.

It involves the implementation, monitoring, and enforcement of security policies that define desired security objectives, rules, and constraints.

1. Policy Definition: Clearly defining comprehensive security policies that cover areas such as access control, authentication, encryption, traffic filtering, incident response, and compliance requirements. 2. Policy Implementation: Configuring security mechanisms and controls, such as firewalls, IDS/IPS, VPNs, ACLs, and encryption protocols, to implement the defined security policies. 3. Policy Monitoring: Continuously monitoring network activity using techniques like log analysis, traffic analysis, system audits, and SIEM systems to ensure adherence to security policies and detect policy violations or anomalies. 4. Policy Enforcement: Taking action to address policy violations by initiating corrective measures, such as blocking malicious traffic, quarantining compromised devices, or alerting relevant personnel. 5. Policy Review and Updates: Regularly reviewing and updating security policies to reflect changes in the threat landscape, technology advancements, or organizational requirements, ensuring their effectiveness and relevance.6. User Education and Awareness: Educating users about security policies and best practices to create a securityconscious culture, reducing the likelihood of inadvertent policy violations and improving overall network security.

The anomaly detector module is the core component responsible for determining whether a flow is normal or abnormal. It analyzes flow information received from the flow collector and classifies flows accordingly. The anomaly mitigator module makes decisions based on the classification provided by the anomaly detector, such as closing connections with nodes or transferring flows.

Overall, effective traffic monitoring, packet inspection, and security policy enforcement are essential for accurate and timely intrusion detection in SDN environments, contributing to network security and mitigating risks.

METHODOLOGY

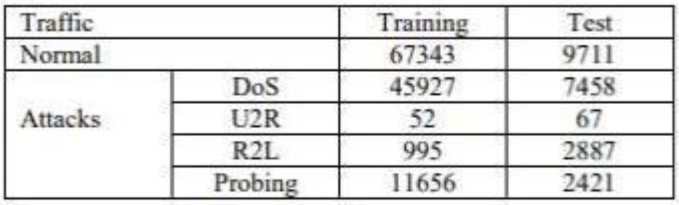
The NSL-KDD dataset is an updated version of the KDD Cup 1999 dataset designed to address issues present in the original dataset and improve the performance of anomaly detectors. It includes records representing normal traffic as well as various types of attacks such as Denial of Service (DoS), U2R, R2L, and probing attacks. The dataset is characterized by 41 features that differentiate each record.

In this study, data preprocessing techniques were applied to derive 118 features from the original 41 features of the NSL-KDD dataset. This preprocessing aimed to enhance the performance of the anomaly detectors being investigated. The dataset was also converted into a binary classification format for ease of implementation, and symbolic features were transformed into numerical values to enable training of the selected anomaly detectors.

In addition to the performance metrics, the effectiveness of the anomaly detectors is assessed based on training and testing time (run time), prediction time for all records in the test set, and the memory size of the anomaly detector. These time and memory considerations provide insights into the practical implementation of anomaly detectors in a Software-Defined Wireless Sensor Network (SDWSN).

Overall, the study focuses on evaluating and comparing the performance of different anomaly detectors using the NSL-KDD dataset. The selected performance metrics and additional considerations such as run time and memory size provide a comprehensive assessment of the anomaly detectors' effectiveness and suitability for an SDWSN environment.

TRAFFIC RECORDS IN THE NSL-KDD DATASET [33]



Traffic		Training	Test
Normal		67343	9711
Attacks	DoS	45927	7458
	U2R	52	67
	R2L	995	2887
	Probing	11656	2421

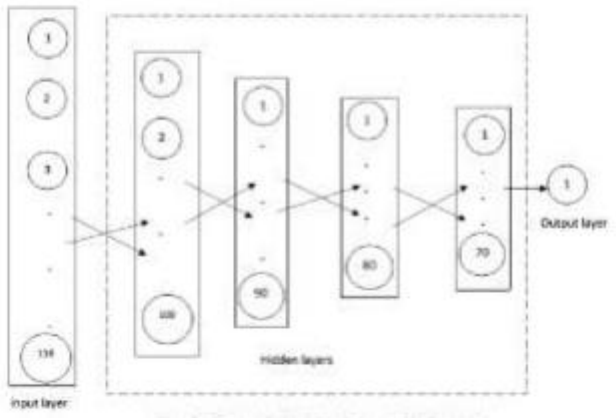
20. K. Ioannis, T. Dimitriou, and F. C. 1-10: Citeseer. (2017, August 10). KDD Dataset 1999 [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/ 21. H.-s. Chae, B.-o. Jo, S.-H. Choi, and T.-k. 184-187, 2013. 22. Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, and M. Sheikhan, "Flow-Based Anomaly Detection Using Neural Network Optimized with GSA Algorithm," in 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, 2013, pp. 76-81. 23. T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," in 2017 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), 2017, pp. 202-206. 24. M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," in Soft computing in industrial applications: Springer, 2011, pp. 293-303. 25. M. Panda, A. Abraham, and M. R., pp. 1-9, 2012. 26. H. F. Eid, M. A. Salama, A. E. Hassanien, and T.-h. Kim, "Bi-layer behavioral-based feature selection approach for network intrusion classification," in International Conference on Security Technology, 2011, pp. 195-203: Springer.
The study implemented three anomaly detectors: a Naive Bayes (NB) method, a Decision Tree (DT) approach, and a Deep Artificial Neural Network (ANN) method.

For the NB-based anomaly detector, the default parameters of the Gaussian NB classifier from the sklearn library were used. The performance evaluation employed 7-fold cross-validation, which yielded the best performance metrics among different k-fold values tested. The recorded metrics for the NB method are presented in Table.

Similarly, the DT-based anomaly detector was implemented using the default parameters of the DT classifier from the sklearn library. The metrics obtained from this approach are listed in Table.

To build the Deep ANN-based anomaly detector, a deep neural network architecture was constructed. The network consisted of an input layer with 118 features, four hidden layers with varying numbers of neurons (100, 90, 80, and 70), and an output layer with a single neuron. The ReLU and sigmoid activation functions were used for the hidden layers and output layer, respectively. The architecture of the deep ANN is illustrated in Fig.

The performance metrics for all anomaly detectors analyzed in the study are summarized in Table VI. Considering SDWSNs with limited controller memory, the NB-based anomaly detector is recommended due to its lower memory consumption. If energy consumption is a primary concern, the NB-based detector is the most suitable choice. However, when memory size is not a constraint, either the DT-based or deep ANN-based anomaly detector can be utilized. Among these options, the DT-based detector exhibits the best overall performance metrics. Furthermore, the DT-based detector has the shortest prediction time, indicating reduced SDWSN latency. On the other hand, the deep ANN-based detector may increase latency the most. Taking these considerations into account, unless specific requirements necessitate a low memory size, a DT-based anomaly detector is suggested as the default choice for implementing an IDS in SDWSNs.



DISCUSSION

Designing and implementing an AI-Based Intrusion Detection System (IDS) for Software-Defined Networks (SDN) involves several important considerations. These include: 1. Data Collection: Collecting relevant network data from sources such as flow records, packet captures, and log files is crucial. Ensure data quality, integrity, and privacy during the collection process. 2. Feature Selection: Determine the relevant features from the collected data for intrusion detection. Carefully selecting features reduces dimensionality and provides meaningful input to the AI models. 3. AI Model Selection: Choose appropriate AI models based on the specific requirements and characteristics of the SDN environment. Machine learning algorithms, deep learning models, or reinforcement learning can be considered. 4. Training Data Preparation: Prepare labeled training data with ground truth labels indicating normal behavior or intrusions. Include diverse network scenarios and attack patterns to enable the AI model to generalize well. 5. Fine-tune hyperparameters and evaluate metrics like accuracy, precision, and recall. 6. Integrate with the SDN controller or network devices for efficient data processing. 7. Response and Mitigation: Define appropriate response actions when an intrusion is detected. Block traffic, raise alerts, notify administrators, or adjust security policies as needed. 8. Performance Optimization: Optimize the IDS's computational resources by employing techniques like parallel processing, model compression, and feature selection. 9. Evaluation and Testing: Conduct rigorous testing to evaluate the IDS's performance, including detection accuracy, false positive rate, and false negative rate. Incorporate feedback and iterate to improve the system over time. 10. Continuous Monitoring and Adaptation: Implement mechanisms for continuous monitoring, model updates, and adaptation to evolving threats. Regularly update and retrain the AI models to stay effective against emerging attack techniques. Additionally, when designing a network security system in general, consider factors such as threat landscape analysis, defense in depth, risk assessment, scalability, user experience, centralized management, compliance requirements, vendor selection, regular updates and patch management, and continuous monitoring with incident response capabilities.

By considering these design considerations and effectively collecting and preprocessing data, organizations can develop robust AI-Based IDSs for SDNs that effectively detect and mitigate intrusions while adapting to evolving threats. The use of AI-Based Intrusion Detection Systems (IDS) in Software-Defined Networks (SDN) has gained attention due to their effectiveness in detecting network intrusions. Several case studies and real-world implementations have demonstrated the practical application of AI-Based IDS in SDN environments. One case study, "Deep Intent," proposed an AI-Based IDS that utilized deep learning techniques to accurately identify various types of attacks in SDN. Another study combined machine learning algorithms and real-time flow monitoring to detect and mitigate network intrusions while minimizing false positives. Additionally, an AI-Driven IDS utilized machine learning techniques to detect and classify network attacks, showcasing high detection accuracy and efficiency for real-time intrusion detection in SDN. Several telecom service providers have also implemented AI-Based IDS in SDN to enhance network security. Open-source IDS projects such as OpenAI-IDS and OpenFlow-IDS provide frameworks and tools for building AI-Based IDS in SDN, with community collaboration for further development. Furthermore, a case study focusing on large-scale SDN implementation outlined the steps involved in developing an AI-Based IDS. It included data collection, preprocessing, AI model selection, training and testing, performance evaluation, deployment, and integration. The benefits of implementing such an IDS included improved intrusion detection, reduced false positives, scalability, enhanced security, and adaptability.

In conclusion, AI-Based IDS for SDN offers valuable capabilities in detecting and mitigating network intrusions. The case studies and real-world implementations discussed provide insights into the practical application and effectiveness of these systems in various SDN environments.

CHALLENGES AND FUTURE DIRECTIONS

The challenges in AI-Based Intrusion Detection Systems (IDS) for Software-Defined Networks (SDN) include adversarial attacks, explain ability, real-time performance, scalability, adaptive learning, integration with threat intelligence, privacy preservation, hybrid approaches, human-in-the-loop, and standardization. Adversarial attacks can deceive IDS systems, while explain ability is important for understanding decision-making. Realtime performance and scalability are required to handle large SDN environments. Adaptive learning and integration with threat intelligence enhance IDS capabilities. Privacy preservation, hybrid approaches, and human-in-the-loop involvement address data privacy, detection accuracy, and human expertise. Standardization and benchmarking facilitate fair comparisons. Future directions include explainable AI, federated learning, adversarial machine learning, self-learning IDS, real-time threat intelligence integration, privacy-preserving IDS, AI-based IDS for IoT-enabled SDN, collaborative threat intelligence sharing, cross-layer IDS approaches, and energy-efficient IDS for SDN. Continued collaboration is crucial for advancing AI-Based IDS for SDN and improving network security.

CONCLUSION

This paper explores the realm of AI-based intrusion detection systems (IDS) for software-defined networks (SDN). It discusses the limitations of traditional IDS in SDN and the need for AI-based IDS. The paper delves into various AI techniques, such as machine learning algorithms, deep learning, reinforcement learning, and hybrid approaches, for intrusion detection in SDN. It addresses the challenges faced by IDS in SDN and provides considerations for designing and implementing AI-based IDS in SDN environments. The paper also discusses evaluation metrics, case studies, and real-world implementations to demonstrate the practicality and effectiveness of AI-based IDS for SDNs. It highlights the ongoing challenges and proposes future research directions. In conclusion, AI-based IDS shows promise in enhancing the security of SDNs, but further research is needed to address challenges and unlock its full potential. Additionally, the paper briefly mentions the performance of anomaly detectors (DT, NB, deep ANN) for interruptions in SDWSNs, suggesting DT-based detectors as the default choice unless specific requirements call for alternative detectors. Deep ANNs are expected to become default detectors for SDWSNs with high-security needs due to their accuracy and ability to identify new attacks as datasets grow.

REFERENCE

1. E. U. Ogbodo, D. Dorrell, and A. M. Abu-Mahfouz, "Cognitive Radio Based Sensor Network in Smart Grid: Architectures, Applications and Communication Technologies," IEEE Access, vol. 5, pp. 19084-19098, 2017. 2. M. Abu-Mahfouz and G. P. Hancke, "Localised information fusion techniques for location discovery in wireless sensor networks," International Journal of Sensor Networks, 21_Publication in refereed journal vol. 26, no. 1, pp. 12-25, 2017. 3. Cheng, J. Zhang, G. P. Hancke, S. Karnouskos, and A. W. Colombo, "Industrial Cyberphysical Systems: Realizing Cloud- Based Big Data Infrastructures," IEEE Industrial Electronics Magazine, vol. 12, no. 1, pp. 25-35, 2017. 4. K. S. E. Phala, A. Kumar, and G. P. Hancke, "Air Quality Monitoring System Based on ISO/IEC/IEEE 21451 Standards," IEEE Sensors Journal, vol. 16, no. 12, pp. 5037-5045, 2016. 5. Cheng, L. Cui, W. Jia, W. Zhao, and P. H. Gerhard, "Multiple Region of Interest Coverage in Camera Sensor Networks for Tele- Intensive Care Units," IEEE Transactions on Industrial Informatics, vol. 12, no. 6, pp. 2331-2341, 2016. 6. M. Abu-Mahfouz and G. P. Hancke, "Evaluating ALWadHA for providing secure localisation for wireless sensor networks," in 2013 Africon, 2013, pp. 1-5. 7. S. W. Pritchard, G. P. Hancke, and A. M. Abu-Mahfouz, "Security in software-defined wireless sensor networks: Threats, challenges and potential solutions," in 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), 2017, pp. 168-173. 8. N. Ntuli and A. Abu-Mahfouz, "A Simple Security Architecture for Smart Water Management System," Procedia Computer Science, vol. 83, pp. 1164-1169, Jan. 2016. 9. Ramotsoela, A. Abu-Mahfouz, and G. P. Hancke, A Survey of Anomaly Detection in Industrial Wireless Sensor Networks with Critical Water System Infrastructure as a Case Study. 2017, p. 2491. 10. H. I. Kobo, G. P. Hancke, and A. M. Abu-Mahfouz, "Towards a distributed control system for software defined Wireless Sensor Networks," in IECON 2017 - 43rd Annual Conference

of the IEEE Industrial Electronics Society, 2017, pp. 6125-6130. 11. H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "Fragmentation-based Distributed Control System for Software Defined Wireless Sensor Networks," IEEE Transactions on Industrial Informatics, pp. 1-1, 2017. 12. M. Ndiaye, P. G. Hancke, and M. A. Abu-Mahfouz, "Software Defined Networking for Improved Wireless Sensor Network Management: A Survey," Sensors, vol. 17, no. 5, 2017. 13. T. Azzabi, H. Farhat, and N. Sahli, "A survey on wireless sensor networks security issues and military specificities," in 2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET), 2017, pp. 66-72. 14. J. Louw, G. Niezen, T. D. Ramotsoela, and A. M. Abu-Mahfouz, "A key distribution scheme using elliptic curve cryptography in wireless sensor networks," in 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), 2016, pp. 1166-1170. 15. S. W. Pritchard, G. P. Hancke, and A. M. Abu-Mahfouz, "Cryptography Methods for Software-Defined Wireless Sensor Networks," in 2017 IEEE 27th International Symposium on Industrial Electronics (ISIE), 2017, pp. 1257-1262. 16. T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for Network Intrusion Detection in Software Defined Networking," in 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), 2016, pp. 258-263. 17. L. Boero, M. Marchese, and S. Zappatore, "Support Vector Machine Meets Software Defined Networking in IDS Domain," in 2017 29th International Teletraffic Congress (ITC 29), 2017, vol. 3, pp. 25-30. 18. M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," International journal of computer science and network security, vol. 7, no. 12, pp. 258-263, 2007. 19. G. MeeraGandhi, K. Appavoo, and S. Srivasta, "Effective network intrusion detection using classifiers decision trees and decision rules," Int. J. Advanced network and application, Vol2, 2010. 20. K. Ioannis, T. Dimitriou, and F. C. Freiling, "Towards intrusion detection in wireless sensor networks," in Proc. of the 13th European Wireless Conference, 2007, pp. 1-10: Citeseer. (2017, August 10). KDD Dataset 1999 [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/ 21. H.-s. Chae, B.-o. Jo, S.-H. Choi, and T.-k. Park, "Feature selection for intrusion detection using NSLKDD," Recent advances in computer science, pp. 184-187, 2013. 22. Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, and M. Sheikhan, "Flow-Based Anomaly Detection Using Neural Network Optimized with GSA Algorithm," in 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, 2013, pp. 76-81. 23. T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," in 2017 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), 2017, pp. 202-206. 24. M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," in Soft computing in industrial applications: Springer, 2011, pp. 293-303. 25. M. Panda, A. Abraham, and M. R. Patra, "A hybrid intelligent approach for network intrusion detection," Procedia Engineering, vol. 30, pp. 1-9, 2012. 26. H. F. Eid, M. A. Salama, A. E. Hassanien, and T.-h. Kim, "Bi-layer behavioral-based feature selection approach for network intrusion classification," in International Conference on Security Technology, 2011, pp. 195-203: Springer.