ISBN 978-989-9121-37-9
28th BARCELONA International Conference on "Innovations in Science, Engineering & Technology" (BISET-24)
April 24-26, 2024 Barcelona (Spain)

Artificial Intelligence based Security Information and Event Management

Raviteja Muvva[1] and Toqeer Israr[2]

[1]Eastern Illinois University, US
[2]Eastern Illinois University, US

Abstract: *Finding an automated method for detecting cyber-attacks is one of the biggest problems in cybersecurity. In this study, we describe an artificial intelligence (AI) method based on artificial neural networks for detecting cyberthreats. The suggested solution uses a deep learning-based detection method to improve cyber-threat identification by breaking down a large volume of recorded security events into individual event profiles. For this project, we created an AI-SIEM system that combines event profiling for data pre-processing with various artificial neural network techniques, such as FCNN, CNN, and LSTM. The system has a strong emphasis on separating true positive warnings from false positive alerts, assisting security analysts in quickly responding to cyber threats. All experiments in this paper on two benchmark datasets (NSLKDD and CICIDS2017) as well as two datasets that were gathered in the real world. We conducted trials utilizing the five traditional machine-learning methods (SVM, k-NN, RF, NB, and DT) to assess the performance comparison with existing approaches. The experimental findings of this study confirm that our proposed methods can be used as learning-based models for network intrusion detection and demonstrate that, when applied in the real world, they outperform traditional machine-learning techniques.*

Keywords: *Refugee Camps` Design, Sustainability Factors, Design Efficiency*

1. Introduction

This research encompasses the design and planning of developing a unique intrusion prevention system (IPS). Our IPS system will perform better in detection fake attacks by using the Machine learning techniques and various algorithms. With the growth of artificial intelligence (AI) capabilities, learning-based systems for identifying cyberattacks have gone further and have achieved notable success in various studies. However, because incidents are always evolving, protecting IT systems from threats and illicit network activity is still exceedingly challenging. Due to numerous network intrusions and illegal activities, effective defenses and security issues were given top priority for finding dependable solutions.

There are typically two main systems for identifying network breaches and cyber threats. The network typically has an intrusion prevention system (IPS) installed, which can primarily use signature-based methods to investigate network protocols and flows. It produces the necessary intrusion alarms, also known as security events, and reports these alerts to another system, like SIEM. The gathering and administration of IPS alerts has been the primary focus of security information and event management (SIEM). Among the different security operations solutions, the SIEM is the most popular and reliable option for analyzing the gathered security events and logs. Additionally, security analysts attempt to evaluate suspicious alerts based on policies and thresholds, as well as to find malicious behavior by examining correlations between events and applying attack-related information. However, because of their high false alarm rates and the vast volume of security data, it is still

challenging to distinguish and detect intrusions against intelligent network threats. As a result, machine learning and artificial intelligence algorithms for identifying attacks have received more attention in the most recent studies. AI advancements can make it easier for security experts to investigate network attacks quickly and automatically. These approaches rely on learning the attack model from previous threat data and employ trained models to find incursions for unidentified cyber threats.

2. Literature Review

Enhanced Network Anomaly Detection Based on Deep Neural Networks

The requirement for data network security has expanded at an outstanding rate throughout the course of recent years because of the dramatic development of Web applications. As a critical protect for the framework of an organization, an interruption identification framework is expected to answer a danger climate that is continually moving. For exact oddity recognition, specialists in ML and data mining have fostered various managed and solo techniques. A subfield of ML known as profound learning utilizes structures looking like neurons for the purpose of learning. Deep learning has fundamentally altered how we approach learning difficulties by making significant advancements in sound handling, PC vision, and regular language handling, to name a few. Examining this fresh out of the plastic innovation for applications in data security is just essential. The reason for this exploration is to decide if inconsistency-based interruption discovery frameworks can be executed utilizing deep learning calculations. Irregularity recognition models in view of autoencoders, convolutional neural networks, and recurrent neural networks were produced with the end goal of this examination. The NSLKDD preparing and test informational collections, NSLKDDTest+ and NSLKDDTest21, were utilized to prepare and assess these profound models. The developers completed all the trials in this work using a GPU-based test seat. Customary ML-based interference acknowledgment models were assembled utilizing wide learning machines, decision trees, random forests, support vector machines, naive bayes, and quadratic discriminant evaluation. Deep and standard ML models were both tried utilizing huge gathering estimates, for example, advantageous working limits, area under curve, precision recall curve, mean normal precision, and arrangement accuracy. The exploratory discoveries of the deep IDS model uncovered promising outcomes for use in obvious idiosyncrasy area systems.

Background

To process extremely huge amounts of data, our suggested system intends to convert many security events into distinct event profiles. By studying a vast quantity of collected data and learning typical and threat patterns while considering how frequently they occur, we are going to build a generalizable security event analysis method. In this work, we recommend a technique for characterizing datasets utilizing basepoints during the data preprocessing stage. When using typical data mining techniques for log analysis, the main problem is frequently the dimensionality of the space, which can be greatly reduced by this approach.

We will test our system's applicability using real IPS security events from a genuine security operations center (SOC), and check its efficacy using performance measures including accuracy, true positive rate (TPR), false positive rate (FPR), and F-measure. Additionally, we will run trials utilizing the five traditional machine-learning methods (SVM, k-NN, RF, NB, and DT) to assess how well they performed in contrast to existing techniques. The two benchmark datasets (NSLKDD and CICIDS2017) that are most often used in network intrusion detection research are also used to evaluate our technique.

Proposed System: We propose an artificial intelligence technique for cyber-threats detection, based on artificial neural networks. The proposed technique converts a multitude of collected security events to individual event profiles and uses a deep learning-based detection method for enhanced cyber-threat detection. For this work, we developed an AI-SIEM (Artificial intelligence-based Security information and event management) system based on a combination of event profiling for data preprocessing and different artificial neural network

methods, including FCNN (Fully connected Neural Network), CNN (Convolutional Neural Network), and LSTM (Long Short-Term Memory Network). The system focuses on discriminating between true positive and false positive alerts, thus helping security analysts to rapidly respond to cyber threats. We will conduct experiments using the five conventional machine-learning methods (SVM (support vector machine), k-NN (K-nearest neighbor), RF (Random Forest), NB (Navie Bayes), and DT (Decision Tree). Consequently, the experimental results of this study will assess if our system is capable of being employed as learning-based models for network intrusion-detection. This proposed system is employed in the real world, and we expect the performance to outperform the conventional machine-learning methods.

Modules

The modules that make up the proposed algorithms are listed below.

1) **Parsing Data:** Data parsing is the process of examining and interpreting unprocessed input or data to draw out relevant information. To do this, the input must be divided into its constituent parts and structural elements, then those parts must be arranged and converted into a format that a computer program or human user can readily understand and use. To produce a raw data event model, this module parses an input dataset.

2) **TF-IDF:** Term Frequency-Inverse Document Frequency is referred to as TF-IDF. It's a metric for quantifying the weight of a phrase in a document in relation to a group of documents that's used in information retrieval and natural language processing. This module will be used to transform raw data into an event vector with normal and attack signatures.

3) **Stage of Event Profiling:** Event profiling is a multidisciplinary approach that incorporates elements of data analysis, natural language processing, machine learning, and domain knowledge. Depending on the type of event, the data that is available, and the objectives of the profiling process, different stages and methodologies may be employed. The processed data will be divided into train and test models based on profiling events.

4) **The Model of a Deep Learning Neural Network:** Using train and test data, this module constructs a training model with the help of CNN and LSTM algorithms. The trained model will be used to produce the prediction score, recall, precision, and F Measure for the test data.

3. Design

Our proposed SIEM system's workflow and architecture are based on artificial intelligence (AI). The data preparation, the learning engine based on artificial neural networks, and the real-time threat detection phase make up the three primary components of our AI-SIEM system. As shown in figure 1, Event profiling, the system's first preprocessing stage, transforms raw data to produce concise inputs for various deep neural networks. The AI-SIEM system performs event profiling, data normalization using the TF-IDF method, and data aggregation with parsing in that order during the data preprocessing phase. The output from each stage is used in the subsequent stages, to produce event data sets, event vectors, and event profiles, respectively. When the system operates on detecting network intrusions in real time, this phase not only comes before the data learning stage but also comes before the conversion of raw security events to the deep-learning engine's input data. Three artificial neural networks are used in the second AI-based learning engine's modeling process.

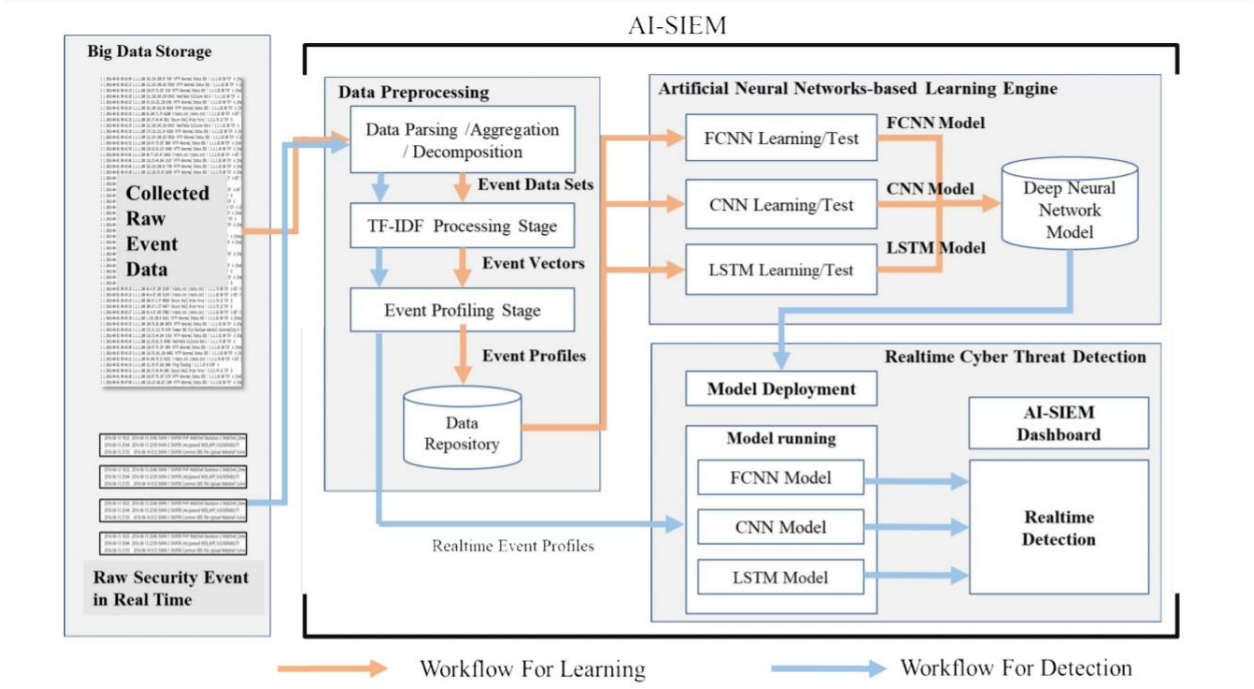

Fig. 1: Workflow diagram

3.1. Data Flow Diagram: For our research project, the user will be authenticated and if the user is not authorized, the flow ends. However, if the user is an authorized user, then the user uploads the training dataset, executes the preprocessing TF-IDF Algorithm, which will generate Event Vector and Neural Network Profile. A Machine Learning model is constructed based on this work and various ML algorithms such as SVM, KNN, Random Forest, Naïve Bayes, Decision Tree are executed. Outputs from these algorithms will be used to build Accuracy, Precision, Recall, and F-Measure Comparison Graphs are constructed.

3.2. Unified Modeling Language (UML) Diagram: The UML is an amalgamation of the best engineering approaches that have been effective in simulating huge, complicated systems. The UML primarily employs graphical notations to convey software project design.

3.3. Proposed Use Case Diagram: In the Unified Modeling Language (UML), a use case diagram is a specific kind of behavioral diagram that results from and is defined by a use-case analysis. The user will pass data and participate as an actor for each phase, where datasets are first uploaded, then preprocessed using the TF-IDF technique to produce an event vector and a neural network profile. By using the inputs from the previous steps, a machine learning model may be constructed in the following step. After that, the following machine learning algorithms will be used: SVM, KNN, Random Forest, Navie Bayes, and Decision tree algorithms. Accuracy, Precision, Recall, and F-measure graphs will be created using the outputs of these algorithms.

3.4. Proposed Class Diagram: User shall pass dataset and executes upload train dataset function. The Upload Train Dataset module reads the uploaded dataset before passing it on to the TF-IDF algorithm to process it. The TF-IDF algorithm will also use system input. Following the execution of the TF-IDF algorithm, event vectors and neural network profiles are also generated. LSTM and CNN models will be created during the neural network profile phase, and these models will be used as input to create machine learning models and send information to the SVN algorithm. Once more, the user will enter the photo and provide input to the SVM algorithm. Following the execution of the algorithms, Accuracy, Precision, Recall, and F-Measure graphs are constructed using the findings from KNN, Random Forest, Navie Bayes, and Decision tree.

3.4. Proposed Object diagram

The user will input the dataset, upload the train dataset, utilize the TF-IDF technique to preprocess the data, build event vectors, and perform neural network profiling before sending the results to the application. Whereas machine learning models will be used on the application side and used as inputs for the algorithms SVM, KNN, Random Forest, Navie Bayes, and Decision Tree. Accuracy, Precision, Recall, and F-measure graphs will be created based on metrics that were obtained from the methods.

3.5. State Diagram

The state diagram will accept data as input and push the data to the next stage, where preprocessing can be done using the TF-IDF algorithm to produce the event tracker that will be given to the neural network profile by developing the machine learning models. The data will then be tested and trained before being passed through various algorithm techniques like SVM, KNN, Random Forest, Navie bayes, and Decision tree algorithms. Finally, comparison graphs based on accuracy, precession, recall, and F-measure comparison graphs would be produced.

3.6. Activity Diagram

In the activity diagram, the system's process flows are depicted. The activity diagram has four phases. Application access is handled during the user phase. The TF-IDF technique will be used to create event vectors and neural network profiles during the dataset.

Stage, where the data will be uploaded and delivered. LSTM and CNN models were developed during this stage. In the third stage, machine learning models that have been trained and evaluated are fed data that is then fed into algorithms like SVM, KNN, Random Forest, Navie Bayes, and Decision Tree. Accuracy, Precision, Recall, and F-Measure graphs will be produced based on the output that was produced by utilizing the algorithms.

3.7. Sequence Diagram

A sequence diagram shows how various system items interact with one another. The fact that a sequence diagram is time-ordered is crucial. This indicates that the precise order of the objects' interactions is displayed step by step. After uploading the datasets to the program, the user uses the TF-IDF method to do pre-processing tasks before creating events and neural networks in the following steps. After developing the machine learning modules, the SVM, KNN, Random Forest, Navie Bayes, and Decision tree algorithms will get the data. Accuracy, Precision, Recall, and F-measure graphs for all the outputs will be returned to the user.

3.8. Collaboration Diagram

The collaboration diagram makes it easier to see every potential interaction between each object and other things. By applying the TF-IDF technique, creating event vectors, and creating neural network profiles, the user will push a dataset for preprocessing. The SVM, KNN, Random Forest, Navie Bayes, and Decision tree methods are used to feed data through machine learning models that are constructed using event vectors and neural network profiles. Accuracy, Precision, Recall, and F Measure comparison graphs were constructed and returned to the user based on metrics produced by algorithms.

3.9. Component diagram

Component diagram the user is responsible for carefully transferring data input and data preparation before sending processed data to the program. The application will develop CNN and LSTM Machine learning models and process data using techniques from Random Forest, KNN, SVM, Navie Bayes, and Decision tree. Accuracy graphs will be produced in accordance with the metrics that were produced.

3.10. Deployment Diagram

The setup of the application's runtime components is depicted in the deployment diagram. User shall take care passing data input and data preprocessing and pass processed data to application, Application will build

CNN and LSTM Machine learning models and process data in Random Forest, KNN, SVM, Navie Bayes and Decision tree algorithms. Depending on the metrics that were generated Accuracy graphs will be generated.

4. Implementation

Algorithms

We employed the following machine learning algorithms in our proposed system.

4.1 Naive Bayes Classifier

Naive Bayes is a classification technique with a notion which defines all features are independent and unrelated to each other. It defines that the status of a specific feature in a class does not affect the status of another feature.

4.2. Decision Tree Classifier

Decision Tree is a supervised machine learning algorithm used to solve classification problems. The main objective of using Decision Tree in this research work is the prediction of target class using decision rule taken from prior data.

4.3. Support Vector Machine (SVM): SVM is one of the standard sets of supervised machine learning models employed in classification. Given a two-class training sample the aim of a support vector machine is to find the best highest-margin separating hyperplane between the two classes.

4.4. K-nearest Neighbor Algorithm: K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. As mentioned in the DFD, we will be using the above-mentioned algorithms to process the datasets.

5. Conclusion

Our method is novel in that it compresses very large-scale data into event profiles and enhances cyber-threat identification using deep learning-based detection algorithms. By reducing false positive signals, it can also help security analysts respond swiftly to cyber threats dispersed across several security events. Two benchmark datasets (NSLKDD, CICIDS2017) and two datasets collected in the real world will be used in a performance comparison to evaluate performance. By comparing our processes to existing methods and using well-known benchmark datasets, we will first show that they may be utilized as one of the learning-based models for network intrusion detection. Second, through the examination of two real datasets, we will present positive results demonstrating that our method also outperformed conventional machine learning algorithms in terms of accurate classifications.

6. Future Work

To address the expanding problem of cyber threats, we will focus on enhancing early threat forecasts using the multiple deep learning approach in the future. To do this, historical data will be examined for long-term trends. Furthermore, to improve the precision of the labelled dataset for supervised learning and provide high-quality learning datasets, many SOC analysts will work directly to record labels of raw security events one by one over a period of months.

7. References

[1] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal,K. Han, "Enhanced Network Anomaly Detection Based on Deep Neural Networks," *IEEE Access*, vol. 6, pp. 48231-48246, 2018. https://doi.org/10.1109/ACCESS.2018.2863036

[2] B. Zhang, G. Hu, Z. Zhou, Y. Zhang, P. Qiao, L. Chang, "Network Intrusion Detection Based on Directed Acyclic Graph and Belief RuleBase", *ETRI Journal*, vol. 39, no. 4, pp. 592-604, Aug. 2017
https://doi.org/10.4218/etrij.17.0116.0305
[3] W. Wang, Y. Sheng and J. Wang, "HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection," *IEEE Access*, vol. 6, no. 99, pp. 1792-1806,2018.
https://doi.org/10.1109/ACCESS.2017.2780250
[4] M. K. Hussein, N. Bin Zainal and A. N. Jaber, "Data security analysis for DDoS defense of cloud-based networks," *2015 IEEE Student Conference on Research and Development (SCOReD)*, KualaLumpur, 2015, pp. 305-310.
https://doi.org/10.1109/SCORED.2015.7449345
[5] S. Sandeep Sekharan, K. Kandasamy, "Profiling SIEM tools and correlation engines for security analytics," *In Proc. Int. Conf.Wireless Com., Signal Proce. and Net. (WiSPNET)*, 2017, pp. 717-721.
https://doi.org/10.1109/WiSPNET.2017.8299855
[6] N.Hubballiand V.Suryanarayanan, ‘‘False alarm minimization techniques in signature-based intrusion detection systems: Asurvey,’’ *Comput. Commun.*, vol. 49, pp. 1-17, Aug. 2014.
https://doi.org/10.1016/j.comcom.2014.04.012
[7] A. Naser, M. A. Majid, M. F. Zolkipli and S. Anwar, "Trusting cloud computing for personal files," *2014 International Conference on Information and Communication Technology Convergence (ICTC)*, Busan, 2014, pp. 488-489.
https://doi.org/10.1109/ICTC.2014.6983188
[8] Y. Shen, E. Mariconti, P. Vervier, and Gianluca Stringhini, "Tiresias: Predicting Security Events Through Deep Learning," *In Proc. ACMCCS 18*, Toronto, Canada, 2018, pp. 592-605.
https://doi.org/10.1145/3243734.3243811
[9] Kyle Soska and Nicolas Christin, "Automatically detecting vulnerable websites before they turn malicious,", *In Proc. USENIXSecurity Symposium.*, San Diego, CA, USA, 2014, pp.625-640.
[10] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, K. Li,"AI2: training a big data machine to defend," *In Proc. IEEEBigDataSecurity HPSC IDS*, New York, NY, USA, 2016, pp. 49-54
https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2016.79
[11] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu and Ali A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," *In Proc. of the Second IEEE Int. Conf. Comp. Int. for Sec. and Def. App*., pp. 53-58,2009.
https://doi.org/10.1109/CISDA.2009.5356528
[12] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization", *Proc. Int. Conf. Inf. Syst. Secure. Privacy*, pp. 108-116, 2018 [online] Available: http://www.takakura.com/Kyoto_data/
https://doi.org/10.5220/0006639801080116
[13] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. TopicsComput. Intell.*, vol. 2, pp. 41-50, Feb. 2018
https://doi.org/10.1109/TETCI.2017.2772792
[14] R. Vinayakumar, Mamoun Alazab, K. P. Soman, P. Poornachandran, Ameer Al-Nemrat and Sitalakshmi Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*,vol. 7, pp. 41525-41550, Apr. 2019.
https://doi.org/10.1109/ACCESS.2019.2895334