CYBER THREAT DETECTION BASED ON ARTIFICIAL NEURAL NETWORKS USING EVENT PROFILES

12.Dr.ARVIND PRASAD

[1.] Professor, Department of Computer Science and Engineering, JNTUH college of engineering nachupally(kondagattu)(JNTUH.CEJ), (.TS). India.
[2.]M.Tech,StudentDepartment of Computer Science and Engineering, JNTUH college of engineering nachupally(kondagattu)(JNTUH CEJ), (.TS).India.

ABSTRACT:

Cyber-threat detection is a major challenge in cyber security because it must be automated and effective. In this paper, we present a cyber-threat detection AI technique based on artificial neural networks. When using a deep learning detection method, a large number of security events can be broken down into individual event profiles, making it easier to detect cyber-threats. Data preprocessing is based on event profiling, and artificial neural networks like FCNN, CNN, and LSTM are used in our AI-SIEM system. True positive and false positive alerts are differentiated by the system for the benefit of security analysts so they can act quickly when confronted with cyber threats. All experiments in this study were carried out on two benchmark datasets (NSLKDD and CICIDS2017) and two real-world datasets. Using conventional machine-learning methods, we conducted experiments and compared the results to those of existing approaches (SVM, k-NN, RF, NB, and DT). We found that our proposed methods outperform traditional machine learning methods in the real world when used as learning-based models for network intrusion detection, as demonstrated by this study's experiments.

Key Words: Cyber security, intrusion detection, network security, artificial intelligence, deep neural networks.

I. INTRODUCTION:

As artificial intelligence (AI) techniques have improved, learning-based approaches to detecting cyber-attacks have improved as well. These approaches have shown significant results in numerous studies. It's still extremely difficult to keep IT systems safe from malicious cyber-attacks because of the constant evolution of these attacks. As a result of various network intrusions and malicious activities, effective defences and security considerations were given top priority for finding reliable solutions. One of two systems is typically used to detect cyber-threats and network intrusions. Network protocols and flows can be checked using signature-based methods by an intrusion prevention system (IPS) on the company's network. The security events are generated by this system, and the alerts generated by it are sent to a different system, such as a SIEM. SIEM has primarily been concerned with monitoring and responding to IPS alarms. When it comes to analysing security events and logs, the SIEM is the most popular and reliable option [5]. Analysts investigate suspicious alerts based on policies and thresholds, as well as identifying malicious behaviour by looking for patterns in the correlation of events and applying attack-related knowledge. They also work to protect against attacks. Detecting intrusions against

intelligent network attacks is still difficult due to high false alarm rates and a large amount of security data. As a result, recent studies on intrusion detection have focused more on machine learning and artificial intelligence techniques for detecting attacks. Artificial intelligence (AI) advancements can assist security analysts in detecting network intrusions more quickly and automatically. There are learning-based approaches that can detect unknown cyber threats by using previously trained models [8] and [9]. A learning-based method geared toward determining whether an attack occurred in a large amount of data can be helpful for analysts who need to instantly analyse numerous events. Those driven by analysts and those driven by machine learning are the two types of information security solutions identified by [10]. Analyst-driven solutions rely on rules that must be followed being determined by security experts known as analysts. Machine learning-driven solutions that look for unusual or unusual patterns can make it easier to detect new cyber threats. Despite this, we discovered four major limitations in the detection of cyber attacks in systems and networks with existing learning-based approaches. Data that has been labelled is required for model training and evaluation before these learning-based detection methods can be used. It's also difficult to collect enough labelled data to train a model accurately. Instead of keeping unlabeled data for unsupervised learning models as is commonly believed, many commercial SIEM solutions do. Second, the learning features that are theoretically used in each study are not real-world features because they are not included in the majority of common network security systems. To make matters more complicated, putting this theory into practise is extremely difficult. Automatic intrusion detection has been automated with the help of deep learning technologies such as those found in the NSLKDD , CICIDS2017 and Kyoto-Honey pot. Prior studies have relied on accurate benchmark datasets, but these aren't useful in the real world because they're missing important features. Real-world datasets must be used to evaluate a used learning model to avoid these drawbacks. An anomaly-based method for detecting network intrusion can identify unknown cyber threats, but false alarms are common [6]. Many false-positive alerts are expensive and time-consuming to investigate. Hackers also use a fourth technique: changing their behaviour patterns over time to make their malicious activities appear to be benign. Assailants' behaviour is constantly changing, so even if learning-based modelling can detect them, it will be ineffective. Almost all of the system's defences have been built to only look at recent network security breaches as a source of threat. Our assumption is that analysing the security event history associated with the generation of events can be one way to detect malicious behaviour in long-term cyber attacks because attacks are constantly evolving. Problems like these are what motivate me. The AI-SIEM system we've developed uses deep learning techniques to distinguish between legitimate alerts and false positives in order to address the problem mentioned above Our system can assist security analysts in quickly responding to cyber threats dispersed across numerous security events. When it comes to resolving this issue, the AI-SIEM system's proposed architecture includes an event pattern extraction technique that aggregates and correlates data based on events with concurrency features. Many deep neural networks could benefit from the input data provided by our event profiles. When compared to historical data for a long time, the analyst is also capable of dealing with all of the data quickly and efficiently. Our research has made the following significant contributions: To handle massive amounts of data,

we'll break down security events into individual event profiles. We developed a generalizable method for security event analysis based on how frequently events occur. This method takes into account both normal and threat patterns. In this study, we used pre-processingbase points to characterise the data sets. Our event profiling approach provides rich input for a variety of deep-learning approaches because it uses artificial intelligence rather than traditional pattern recognition techniques. • With our artificial intelligence event profiling approach, we significantly reduce the dimensionality of the space we're working in. Since conventional machine-learning methods produce more false alarms, our approach can help reduce the number of false positives and thus reduce the number of alerts that analysts receive. Using real-world IPS security events from an actual security operations centre (SOC), we test the applicability of our system and validate it using performance metrics such as accuracy, true positive rate (TPR), false positive rate (FPR), and F-measure. Our experiments compared the performance of the five most common machine-learning approaches to previous approaches (SVM, k-NN, RF, NB and DT). For the purpose of network intrusion detection, we also put our method through its paces on two widely used benchmark datasets (NSLKDD and CICIDS 2017). We broke down a large amount of data into discrete event occurrence profiles using the TF-IDF mechanism. To generate the event profiles, we also compute the similarity value between each TF-IDF event set and the designated baselines. For models like FCNN, CNN, and LSTM, AI-SIEM uses the generated event profiles as input. These models run on top of each other in the input layer. Since our system can protect IT systems from cyber threats, we plan to demonstrate its applicability using two well-known benchmark data sets and two real data sets obtained from running an IPS. Even though the NSLKDD and CICIDS2017 datasets have drawbacks, they are still widely used to compare machine-learning methodologies for assessment. The findings are compared with those of other researchers using real datasets and two benchmarks. There have been promising results in test datasets for machine-learning approaches, but they must also perform well when applied to actual data.

II. PRELIMINARIES:

This section provides a quick review of the study's context. IDS/IPS and SIEM are introduced first,

For starters, IDS/IPS and SIEM.

With the explosion of data and the internet, an IPS has become a must-have system for virtually any organisation or industry. Intelligent network attacks continue to exist today, however, and an IPS system's detection and response capabilities are limited. This is due to the fact that signature-based detection, rather than anomaly detection, is what they rely on the most. Meanwhile, new intrusion methods such as quick attacks are becoming more common [6]. There is a high false positive rate in the majority of IPS solutions, making it difficult to identify new or unknown attacks. An IPS is also limited by six other factors, such as the volume, accuracy, diversity, dynamic nature of an attack and adaptability. These factors were discussed in depth. Because of these restrictions, a SOC security analyst's ability to make precise decisions is severely hampered.

SIEM (System Information and Event Management):

A SIEM has been recognised as an important enterprise network and security infrastructure component, providing an overall view of security management with a focus on enterprise information

technology (IT) security. With the help of SIEM, a company's environment can be protected from cyber threats by matching patterns in the collected data.] Network security systems (such as firewalls and IDS/IPS) send logs and alerts to the SIEM system, which can then be consolidated and analysed in detail. When analysing SIEM's IDS/IPS alerts (security events), the analyst uses predefined security policies and thresholds to find cyber-attacks. To find consolidated malicious behaviour, they also perform correlation analyses on security events and relevant situations based on previously known patterns of cyber threats. Because they are continuously generated by various network security systems, these heterogeneous events have a wide distribution (such as IPSs and FWs). There are therefore difficulties in distinguishing between real and false positive alerts in traditional policy-based threat detection systems. Aside from being difficult and expensive in theory, in practise this method of analysis has proven to be impractical. When it comes to cyber-threat detection, SIEM analysts put in a lot of time and effort to tell the difference between real and fake security alerts in collected events. In recent years, SIEM development has placed a high priority on applying machine learning and artificial intelligence (AI)-learning techniques, which is referred to as AI-based SIEM. There are still several challenges for an AI-based SIEM despite the fact that applying these techniques has decreased the amount of human labour required. Significant drawbacks, such as (1) the need for a high level of analyst interaction, (2) the absence of labelled data and (3) attacks that are constantly evolving have already been mentioned.

B. DEEP LEARNING TECHNIQUES:

Beyond machine learning, neurons are being used as mathematical structures that are similar to human neural networks, with great success in recent years thanks to deep learning advancements. The most widely used deep learning models are convolutional and recurrent neural networks. For learning spatial features like image processing, RNNs are better suited than CNNs for learning from time-continuously differentiable features of data.CNNs are data-processing architectures that excel at handling spatial data. CNNs are used in a wide range of applications due to their understanding of the input's partial specificity, local characteristic, and shared parameter scheme. CNNs have already produced impressive results in a variety of fields, including

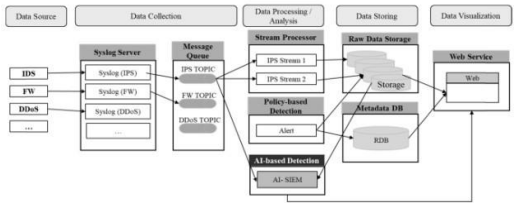

Figure1. The design of our AI-based SIEM big data platform

Medical text analysis, image classification, and malware classification are just a few examples of cutting-edge research in these fields. A number of studies have demonstrated the viability of using CNN for network intrusion detection by identifying malicious events, networking flow, and network connections A recurrent structure can learn from the data's sequence information by repeating itself over and over again. RNN and LSTM are well-known recurrent structures. In comparison to RNNs, LSTMs have a unique recurrent architecture designed to improve storage capacity. In part, this is due to the RNN's limited ability to retain past input information for long enough to accurately model the input sequence's long-term structure. As a result, the forget gate is a part of LSTM networks. In areas like speech recognition and machine translation, LSTM's ability to learn long sequence

data has enabled successful empirical results [3], [10].

C. BIG DATA PLATFORM:

Long-term security log collection and storage on a big data platform are standard practises. Also, the big data platform can detect and respond to cyber threats. As a result, historical data from the platform can be used to investigate and combat cyber threats. Our big data platform is based on distributed computing technologies and can handle large amounts of data while remaining scalable, especially when working with security event logs. The big data platform's system architecture is shown in Figure 1. The platform consists primarily of a data collection, processing, analysis, and storage system for analysing cyber-threat information using long-term security data. This platform is capable of continuously collecting and processing the large amount of security events that are streamed in. AI-based SIEM on the big data platform can be used in conjunction with our approaches. Artificial intelligence (AI) techniques applied to the platform in this study can tell the difference between true and false alerts in the real world.

III. RELATED WORKS:

This section discusses deep learning-based intrusion detection as well as real-time security event analysis research. In the last few years, many studies in the field of cybersecurity have focused on artificial intelligence-based intrusion detection and various AI- and machine learning-based techniques have been proposed to improve cyber-threat detection. The datasets used are still restricted to NSLKDD, despite the fact that artificial intelligence and machine learning-based techniques have produced significant results in these studies. In contrast to this study, others [8, 10] used real-world security events and logs. As far as addressing the aforementioned issues, these studies are more in line with what we've done. Du et al., Liao and Vemuri and Zhang et al. have all used our method.

A. DEEP LEARNING-BASED INTRUSION DETECTION:

Naseer et al. [1] have developed models for intrusion detection that use a variety of deep neural network architectures, such as CNNs, Auto encoders, and RNNs. These models were trained on NSLKDD's training and test datasets. DCNN and LSTM models had an accuracy of 85% and 89%, respectively, on the test dataset. Based on their research, Zhang et al. [2] identified two types of network intrusion detection approaches: single-algorithm direct detection and multi-method detection. The author proposed a new detection model based on directed acyclic graphs and belief rule bases (BRB). In comparison to conventional detection models, the DAG-BRB model with KDD 99 dataset had a higher detection rate. Wang et al. [3] have developed a network traffic feature learning system using a hierarchical spatial and temporal intrusion detection system (HAST-IDS). LSTM networks learn the temporal characteristics of network traffic after learning the spatial characteristics with deep convolutional neural networks (CNNs). The experiments were carried out by DARPA and ISCX datasets.

B. REAL SECURITY EVENT ANALYSIS:

For the purpose of predicting security events with deep learning, Shen et al. developed Tiresias. According to the research, RNNs can make predictions about a machine's future behaviour based on past data. Tested on a commercial IPS's 3.4 billion security events, the approach succeeded in accurately predicting the next machine event

with a precision of up to 0.93, according to the results. On top of all that, the system was extremely precise in a difficult situation, and it also consistently produced good outcomes. Existing systems cannot predict cyber-attacks as well as new machine learning techniques developed by Veeramachaneni and colleagues [10]. These new techniques continuously incorporate human expert input. After using a ranked metric to label data for several months, the analyst then fed the labelled data into the supervised learning module to see if an attack occurred in the future. This method is roughly three times better than previous benchmarks, while also reducing the number of false positives by a factor of five. It uses six anomaly detection methods to detect 85% of attacks. Data from millions of users was used to test the system, which worked with 3.6 billion "log lines" of data over a three-month period. Anomaly detection using hybrid auto-encoder approaches has recently been proposed. Deep Log, a deep neural network model utilising LSTM, was proposed by Liao and Vemuri to train a system's log patterns (such as log key patterns and corresponding parameter value patterns). Using log key and parameter value anomaly detection models and the term frequency inverse document frequency (TF-IDF) vector, this study seeks to find anomalous log entries. A study by the author found that Deep Log outperformed existing log-based anomaly detection methods, with an F-measure of 96% for HDFS data and an F-measure of 98% for Open Stack. For early-stage enterprise infection detection, Oprea et al. used DNS logs. A new framework based on graph theory-inspired belief propagation was put forth by the authors. They demonstrated the efficacy of their methods on two large datasets. The authors were able to achieve impressive precision by using DNS logs spanning two months. On 38 terabytes of web proxy logs amassed at the border of a large company, these algorithms are applied. After the "hints" data was collected from the security analysts at the SOC, the final product was built using it. Using streaming console logs, Zhang et al proposed an innovative system for detecting early warning signals of IT system failures. Text mining techniques such as TF-IDF and LSTM were used to automate the system's training process with labelled data, and the results were impressive. Researchers found that the proposed method outperformed current machine learning approaches at predicting complex IT failures.

IV. SYSTEM OVERVIEW:

Using artificial intelligence to detect cyberthreats is described in detail in this section of the proposed AI-SIEM system's architecture. The AI-SIEM system incorporates a data preprocessing mechanism in addition to deep learning techniques for dealing with extremely large network events. In order to detect cyber-threats automatically, the AI-main SIEM aims to conduct multiple analyses on network security events related to real alerts.

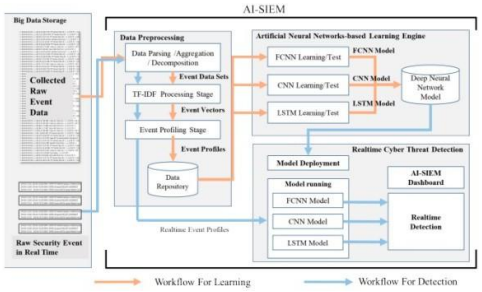

Figure 2: The AI-based SIEM system's work flow and architecture

Engines. Using multiple cores of a GPU speeds up the analysis by using parallel processing power. Figure 2 depicts the workflow and architecture of this AI-based SIEM system. An artificial neural network-based learning engine and real-time threat detection are the three main phases of the AI SIEM system. Raw data is transformed into condensed

inputs for a variety of deep neural networks using event profiling, the system's first preprocessing step. To begin, the AI-SIEM system aggregates data using parsing, normalises data using the TF-IDF mechanism, and profiles events. As shown in Figure 2, the output from each stage is used to generate event data sets, event vectors, and event profiles in the following stage. The system must detect network intrusions in real time before moving on to the data learning stage, and the raw security events must then be converted into the input data for the deep-learning engine. There are three AI-based learning engines that use artificial neural networks. During the data learning stage, the preprocessed data is fed into three artificial neural networks (ANNs), and each ANN learns to find the most accurate model therein. For the final step in real-time threat detection, each ANN model uses the trained model to mechanically classify each security raw event, and the dashboard only reveals to security analysts the recognised true alerts for reducing false ones. ANNs are used for the second phase of learning from the data in Section VI, which covers each step in the preprocessing process in Section V.

V. METHODOLOGY:

A preprocessing method called event profiling will be discussed in this section. The procedure includes data aggregation and decomposition, TF-IDF normalisation, and the creation of an event profile. So let's get started by constructing an event set from scratch. A detailed explanation of TF-IDF-based event vectorization will follow that. Finally, we show how an event profiling approach can be used to profile inputs into three deep learning models. The discovery that concurrent event sets can profile raw event data inspired the method's design in large part. Figure 2 illustrates how the AI engine's preprocessing is carried out by sequentially combining each of the methods shown:

A. DATA AGGRERATION AND DECOMPOSITION:

Much of the event data can be distilled by using a profiling method to identify patterns among the many observations. We needed a way to deal with large amounts of real-world streaming event data, so we created statistical event sets. According to our method's foundation, other events that occur simultaneously with ours can be gleaned of information about their occurrence. Each event is mapped into a single event set using the big data platform's sliding window and a predefined interval, which can then be configured to be part of overlapping sets. To put it another way, the sliding window allows multiple profiles on a single log to be stacked on top of each other. There is no sequence in this case because the number of concurrency event name types in each event set is deterministic for true-positive events rather than using a concurrency-based pattern [8]. This is primarily because unexpected circumstances can slightly alter the course of events. The ordering of the two sequences is clearly different, but the event occurrences are the same in both sequences a and b. This is an excellent example. However, in reality,

Because system processes, resources, and the network can all affect IPS' sequence, we use a concurrency-based method that relies on co-occurrence information. Even though this method does not allow calibration of the variable sequence gap, it is still more accurate than the sequence itself. Using the source and destination addresses Si and Dj, an event set is created for each time interval slide using the raw event data sets EST=t i,j. Because of this, our system generates multiple event data sets for a given period of time.

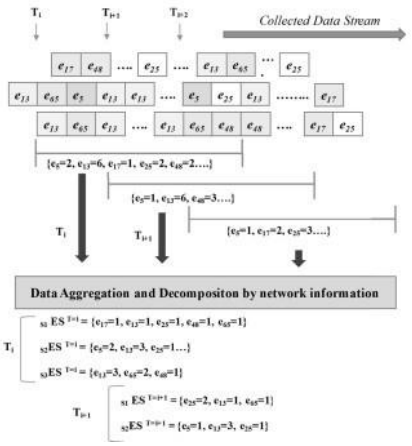

Figure 3. Sliding window for data aggregation and decomposition based on source and destination addresses

Figure 3 uses sliding windows to show the aggregation and decomposition of data. Connection unit decomposes the event set e5 to give us the following values: S1 ES is equal to 1 and S3ES is equal to 3 in the case of the first window Ti, and S3ES is equal to 3 in the case of the second window Ti (see Figure 1). This process is ongoing and is powered by the acquired knowledge.

B. TF-IDF DATA NORMALIZATION:

This subsection uses the frequency of unique event names found in event sets like event set ESi to create a representation for the learning algorithm and classifiers. Vector space models are commonly used in information retrieval for document representations. Our goal is to incorporate this technique into a model for detecting intrusions. This can be done by identifying the IPS pattern and then transforming each event set into Table: There are a variety of symbols and notations used.

Terms	TF-IDF For Common Text Categorization	TF-IDF for Malware Detection in Liao *et al.* [39]	Substitution of TF-IDF for our System
m	total number of documents	total number of processes	total number of event sets
n	total number of distinct words	total number of distinct system calls	total number of unique event names
n_i	number of times i th word occurs	number of times i th system call was issued	number of times i th event was issued
tf_{ij}	frequency of i th word in the j th document	frequency of i th system call in process j	frequency of i th event in event set j
D_j	j th training document	j th training process	j th training event set
X	test document	test process	test event set

a path to follow. Vector space proximity is assumed for event sets in the same concurrency. Thus, as can be seen in Table 1, we use the vector space model to categorise texts by substituting different threat detection factors. Each set of events in the model is represented by a vector of actual events that took place within that set. In the training dataset, m represents the number of rows and n the number of name types for events. To represent the occurrence of an event in an event set, each entry has the formula E = (eij), where ej is the weight of event j in set i. An m-by-n matrix E is what we're dealing with here. There are numerous ways to figure out how much weight eij should be assigned to something. It's possible that dataset A contains tfij unique event frequencies and nj unique names because dataset A has an odd number of named events. This means that dataset A has m unique names and an odd number of named events, and dataset A has an odd number of named events. Documents are weighed using the TF-IDF approach, which uses the term frequency-inverse document frequency (eij=tfij) instead of a simple Boolean weighting (eij=tfij).

$$e_{ij} = tf_{ij} \times \log\left(\frac{m}{n_j}\right) \qquad (1)$$

TF-IDF is a statistical technique for indexing terms based on their importance because it uses vectors to represent both term frequency and term presence. If an event occurs frequently, the numerical value will

be low, while an extremely rare event will have a high numerical value, as an example.. The number of event sets in your data set can be used to generate lengths for each column and row in a matrix A created with TF-IDF. Matrix A is made up of event vectors.The frequency of system calls invoked during the execution time of a programme to detect malicious activities using the TF-IDF for learning programme behaviour. Replacement of the TFF (transfer function) in our AI-SIEM system is shown in Table 1. Assume that aij is the dataset's TF-IDF value for the ith row and jth column. Specifically, we'd like to create a mapping F:E EP for deep learning, where EP represents the event profile dataset for the entity Ei = E1, E2, E3, and so on and so forth. As a result, our dataset contains m rows and n categories, all of which are represented numerically. The number of columns in the collection determines the size of the TF-IDF event set vector, which is dependent on the type of event that occurred. There are countless variations, so reducing over fitting due to a large dimension is essential.

VI. EXPERIMENTS AND RESULTS:

This section contains two benchmark datasets as well as two real datasets that we have gathered over the course of research. We'll start with a discussion of the testbed's physical environment. The experiment's metric will be discussed after that. Our performance evaluation is constantly compared between SVD and traditional machine learning methods. In Subsection E, we discuss the experimental findings in depth, and in the final section of the paper, we show the system we recommended be implemented.

A. TEST EVIRONMENTS:

In order to carry out performance evaluations, a specialised testbed was created just for that purpose. This testbed is made up of two components: a big data platform and an AI-SIEM system. It had also been collecting real-world IPS data for several months during this time. The dataset was built using performance evaluation data and refined with minor data filtering. Security event formats vary from device to device and vendor to vendor, but most events always contain timestamp, source IP address, destination IP address and port details, protocol and flow details as well as rule names. Traditional SIEM stores security events in a standard format with minor additions such as data tagging and enrichment. ESX-1 and ESX-2 data sets contain a variety of IDS/IPS types, so they can be used with other SIEMs and SOCs with ease. IPS sensors aren't available commercially, so we created an emulator that can be used in a variety of scenarios. Once it's finished reading and synthesising a security event dataset, the AI-SIEM system then uses the syslog protocol to send a syslog packet to the system. For the two benchmark datasets, the sensor emulator takes data from the local system and sends it to the AI-SIEM system. Our EP-ANN from AI-SIEM was implemented using TensorFlow . With 128GB of memory and 2.5 GHz processors, the EP-ANN methods were tested. Two Nvidia Tesla P100 GPUs serve as the system's accelerator.

B. METRICS AND EXPERIMENTAL SETUP:

There are a total of four variables to take into consideration. In order to assess the system's overall performance, metrics such as accuracy, TPR, FPR, and F-measure are commonly used. When evaluating a system's TPR, consider how well it detects threats based on its past performance. Data that has been incorrectly classified can be evaluated with the help of FPR, a statistical tool. To find out how accurate the attack detection system is, divide Precision by the number of True Positives (TP) and False Positives (FP) in

the data. Precision equals the percentage of true attacks among all classified attacks. When calculating precision, divide the total number of classified attacks by the total number of true attacks. TN is defined as the presence of more normal data than attack data (True Negative). FN is assigned when there is an abnormally high amount of attack data (False Negative). Accuracy, precision ratio (PRR), time to replacement rate (TPR), and F-measure are all defined in the section below.

$$TPR(Recall) = \frac{TP}{TP + FN} \quad (13)$$

$$FPR = \frac{FP}{TN + FP} \quad (14)$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (15)$$

$$F - measure = \frac{2 \cdot Precison \cdot Recall}{Precision + Recall} \quad (16)$$

The ROC (Rapid Oscillation Cap We compare the quality of detection performance using a receiver operating characteristic (ROC) curve and an area under the curve (AUC) value. It shows how often a false positive result occurs in binary classifiers, as well as how often the correct result occurs (TPR). Data points classified incorrectly as being under attack are counted as part of the FPR. For the purposes of debating TPR, keep in mind that it refers to the percentage of correctly predicted attack data points, not the overall percentage of attack data. The ROC curve dictates that sensitivity and false-positive rate (FPR) must be equal. As the ROC curve approaches the top-left border and vice versa, prediction quality improves [1]. To put it another way, AUC measures how well a binary classifier can predict label values based on input data. Classifiers with an AUC value greater than or equal to 1 are considered to be more accurate. The classifier performs poorly if the AUC is less than 0.5 [1].

SYSTEM DEPLOYMENT:

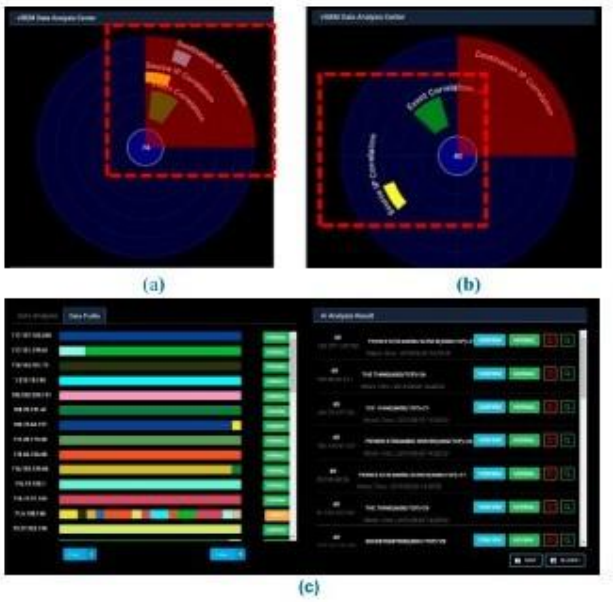

Figure 4: Images taken from the AI-based SIEM system's dashboard for in-the-moment surveillance. Visualization of threat detection (a) and normal state (b) Event profiles and cyber threat lists can be seen in this view..

VII. CONCLUSION:

This paper introduces the AI-SIEM system, which utilises event profiles and artificial neural networks. Deep learning-based detection methods are unique in that they use large amounts of data to condense them into event profiles, which improve cyber-threat detection. The AI-SIEM system gives security analysts the ability to respond quickly and efficiently to major security alerts by comparing historical security data with the system. A reduction in false positive alerts will better equip security analysts to deal with cyber threats dispersed across many different types of security events. To reach our conclusions, we compared the results of two benchmark datasets (NSLKDD and CICIDS2017) with two real-world datasets. A comparison experiment with well-known benchmark datasets showed that our mechanisms can be used as one of the learning-based models for

network intrusion detection. With respect to classification accuracy, the second set of results showed that our technology outperformed conventional machine learning methods. This showed promise when tested on real datasets. As the problem of cyber attacks evolves, future research will concentrate on improving earlier threat predictions through a variety of deep learning approaches that uncover long-term patterns in historical data. Many SOC analysts use supervised learning to improve the precision of labelled datasets and build good learning datasets by recording raw security events one by one over several months

VIII. REFERENCES:

[1] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, ‘‘Enhanced network anomaly detection based on deep neural networks,’’ IEEE Access, vol. 6, pp. 48231–48246, 2018.

[2] B.-C. Zhang, G.-Y. Hu, Z.-J. Zhou, Y.-M. Zhang, P.-L. Qiao, and L.-L. Chang, ‘‘Network intrusion detection based on directed acyclic graph and belief rule base,’’ Electron. Telecommun. Res. Inst. J., vol. 39, no. 4, pp. 592–604, Aug. 2017.

[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, pp. 1792–1806, 2018.

[4] M. K. Hussein, N. Bin Zainal, and A. N. Jaber, ‘‘Data security analysis for DDoS defense of cloud based networks,’’ in Proc. IEEE Student Conf. Res. Develop. (SCOReD), Kuala Lumpur, Malaysia, Dec. 2015, pp. 305–310.

[5] S. S. Sekharan and K. Kandasamy, ‘‘Profiling SIEM tools and correlation engines for security analytics,’’ in Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET), Mar. 2017, pp. 717–721.

[6] N. Hubballi and V. Suryanarayanan, ‘‘False alarm minimization techniques in signature-based intrusion detection systems: A survey,’’ Comput. Commun., vol. 49, p. 1Â17, Aug. 2014.

[7] A. Naser, M. A. Majid, M. F. Zolkipli, and S. Anwar, ‘‘Trusting cloud computing for personal files,’’ in Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC), Busan, South Korea, Oct. 2014, pp. 488–489.

[8] Y. Shen, E. Mariconti, P. A. Vervier, and G. Stringhini, ‘‘Tiresias: Predicting security events through deep learning,’’ in Proc. ACM CCS, Toronto, ON, Canada, Oct. 2018, pp. 592–605.

[9] K. Soska and N. Christin, ‘‘automatically detecting vulnerable Websites before they turn malicious,’’ in Proc. USENIX Secur. Symp., San Diego, CA, USA, 2014, pp. 625–640.

[10] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li, ‘‘AI2 : Training a big data machine to defend,’’ in Proc. IEEE BigDataSecurity HPSC IDS, New York, NY, USA, Apr. 2016, pp. 49–54.