IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 July 2024, accepted 12 August 2024, date of publication 15 August 2024, date of current version 27 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3444703

APPLIED RESEARCH

SSOLV: Real-Time AI/ML-Based Cybersecurity via Statistical Analysis

MAKIA S. POWELL, (Member, IEEE), AND BENJAMIN M. DROZDENKO, (Member, IEEE)
Naval Undersea Warfare Center Division Newport (NUWCDIVNPT), Newport, RI 02840, USA

Corresponding author: Makia S. Powell (makia.s.powell.civ@us.navy.mil)

This work was supported in part by the Naval Undersea Warfare Center Division Newport (NUWCDIVNPT), and in part by the Office of Naval Research under Grant N0001422WX01637.

ABSTRACT The next generation architectures of computer networks and systems and commercial technologies such as Big Data, Decentralized Storage, and 6G require novel approaches to prevent cybersecurity breaches which can negatively affect organizations, operations, and individual customers and stakeholders. In this proposal, we present an approach for identifying transformed features via statistical analysis which can be used in Artificial Intelligence (AI) and machine learning (ML) based systems. We also present a deep learning framework, Small Set of Linearized Variables (SSOLV), for training neural networks based on labeled Zeek datasets containing both malicious and benign activity in real or near-real time. In addition, we present a mechanism for transfer learning using domain adaptation techniques to adapt neural networks trained on one labeled Zeek dataset to another neural network trained on a different labeled Zeek dataset. This research uses a combination of 3 techniques commonly used in network traffic flow analysis: deep neural networks, linear regression, and ANOVA. Our results show that we can classify malicious activity with up to 97-99% accuracy in select cases and high precision (>95%) and recall (>90%) rate. This framework demonstrates a mechanism that stand-alone systems disconnected from larger networks can use to recognize adversarial activity in real time and is transferrable to other stand-alone systems. This work is patent pending under U.S. Patent App. Ser. No. 18/121,716 "Linearized Real-Time Network Intrusion Detection System" and U.S. Patent App. Ser. No. 17/900,982 "Real-Time Network Intrusion Detection System."

INDEX TERMS Artificial intelligence, machine learning, cybersecurity, real-time, statistics, ANOVA, linear regression, artificial neural networks, KDDCUP99, KDD99, UNSW-NB15, CICIDS2017.

I. INTRODUCTION

The next generation architectures of computer networks and systems and commercial technologies such as Big Data, Decentralized Storage, and 6G require novel approaches to prevent cybersecurity breaches which can negatively affect organizations, operations, and individual customers and stakeholders. According to a recent 2022 IEEE Industry Standards Report, as of 2021 the average cost of an industry cybersecurity breach is $4.24 million US dollars, the number of current major breaches increase yearly, and over 90% of healthcare organization have reported a security breach within the past three years [1]. These breaches have both individual and organizational consequences and can leave customers at increased risk of identity theft, result in loss of data and operations, and erode organizational confidence. These losses can even affect the national supply chain as in the case of the Colonial Pipeline incident. The importance of securing organizational computer systems has risen in so many countries that in September 2022, the United States White House issued a memorandum which covers enhancements for increasing the security of the national software supply chain [1]. The cybersecurity of computer systems has become a major concern; in particular, there is an increased need for network traffic analysis and detecting the presence of threat actors which include Zero Day, or new unpublished threats. In industry, commercial intrusion detection and prevention systems (IDSs/IPSs) have been useful for protecting systems on the general internet for existing documented threats, but they are lacking in that they

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaojun Steven Li.

cannot detect Zero-day or previously unpublished cyber threats. To accommodate large amounts of traffic, flow analysis using such technologies as the Zeek network security monitor have shown clear advantage for real-time concerns since the flows can be rendered in real time or near real time. Whereas deep neural networks would take considerable time to process raw network PCAP files, training a deep neural network on Zeek flow data would show significant time improvement in recognizing modern cyberattacks.

Recent advancements in artificial intelligence and machine learning have made it possible to detect anomalies in network traffic that indicate cyberattacks in real time using Time-Series Analysis [2], using popular network monitoring tools such as Zeek (formerly Bro-IDS) or CICFlowmeter [3], [4], [5], and include Zero Day or novel exploit detection using Deep Learning networks [6]. These new artificial intelligence and machine learning methods can be integrated into a novel architecture known as a Zero Trust architecture as part of its Intrusion Detection System (IDS) to proactively detect and act against cybersecurity breaches before they traverse the network [7]. Proactively preventing cybersecurity breaches using these novel techniques can result in financial benefits, improved operations, improved data protection, and improved customer experience.

In this proposal, we present an approach for identifying transformed features via statistical analysis which can be used in Artificial intelligence and machine learning based systems. We also present a deep learning framework for training neural networks based on labeled Zeek datasets containing both malicious and benign activity in real or near-real time. In addition, we present a mechanism for transfer learning using domain adaptation techniques to adapt neural networks trained on one labeled Zeek dataset to another neural network trained on a different labeled Zeek dataset. Our results show that we can classify malicious activity with up to 97-99% accuracy in select cases and high precision (>95%) and recall (>90%) rate. This framework demonstrates a mechanism that stand-alone systems disconnected from larger networks can use to recognize adversarial activity in real time and is transferrable to other stand-alone systems.

II. RELATED WORK

Much of the work that has been done for AI/ML based cybersecurity has focused on a single dataset and is based upon the network flows for a single network. For example, the popular KDDCUP99 dataset benchmark features normal traffic and multiple classes of cyberattack but suffers from duplicates which affects the dataset distribution and accuracy and does not address modern cyberattacks such as malware [8]. The duplicates can negatively affect the Statistical Conclusion Validity of the dataset since they can change the distribution of the data, making statistical inferences derived from the data less valid [9]. Unlike [9], which focuses on using statistics in general, this research focuses on statistical analysis for cybersecurity applications. This research also uses the KDDCUP99, and newer UNSW-NB15 and CICIDS2017 datasets which remove duplicates and addresses modern cyberattacks.

The popularly cited UNSW-NB15 dataset and its feature analysis represents an improvement with normal traffic and 8 different classes of modern cyberattacks including malware. The dataset also uses the popular Zeek network traffic analysis tool to generate flows for analysis. However, the dataset deals with over 30 features, much of them generated post network capture, which negates real-time network traffic analysis [4]. Statistical analysis was done by the dataset authors on the UNSW-NB15 Dataset but still resulted in a reduced set of features which included customized features which negate real-time analysis [8]. This research uses two base features which are available in real-time network flows such as Zeek and derived an additional four features that can be generated in real-time for AI/ML real-time network anomaly detection.

Statistical analysis with methods such as ANOVA to derive features has been accomplished for different tasks such as Common Spatial Pattern analysis for brain-computer interfaces [10]. However, there is a lack of research in using similar techniques to derive features for network anomaly detection cybersecurity. This research uses ANOVA and statistical analysis to derive a set of two base features which are available in real-time network flows such as Zeek and derived an additional four features in real-time for AI/ML real-time network anomaly detection.

III. BACKGROUND

This research uses a combination of 3 techniques commonly used in network traffic flow analysis: deep neural networks, linear regression, and ANOVA.

A. DEEP NEURAL NETWORKS

Deep neural networks (DNNs) are a category of artificial neural networks (ANNs). Deep learning has already transformed traditional internet businesses like web search and advertising and is also enabling brand new products and businesses and ways of helping people to be created. It enables better healthcare, including improvements in reading X-ray images, delivering personalized education, precision agriculture, self-driving cars and many other applications. ANNs have a graph representation as shown in Fig. 1.

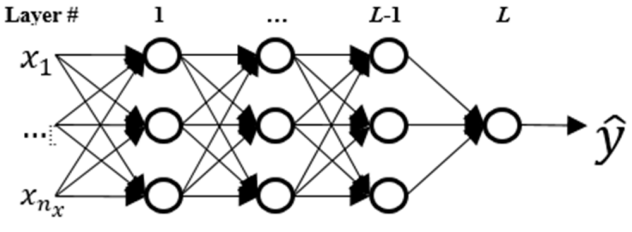

FIGURE 1. Deep neural network graph representation.

For an ANN to be a deep neural network, the number of layers L is very large and characterized by a very high number of hidden units. In the era of big data, it has been generally found that there are functions you can compute with a "small" L-layer deep neural network that shallower networks require exponentially more hidden units to compute. The values of predicted outputs given the inputs are computed in forward propagation, which is given by equation 1.

$$\begin{aligned} Z^{[1]} &= W^{[1]}X + b^{[1]} \\ A^{[1]} &= g^{[1]}\left(Z^{[1]}\right) \\ Z^{[2]} &= W^{[2]}A^{[1]} + b^{[2]} \\ A^{[2]} &= g^{[2]}\left(Z^{[2]}\right) \\ &\cdots \\ A^{[L]} &= g^{[L]}\left(Z^{[L]}\right) = Y \end{aligned} \tag{1}$$

where X is the $\mathbb{R}^{n_x \times m}$ input matrix, $W^{[l]}$ is the weight matrix at layer l, $b^{[l]}$ is the bias vector at layer l, $g^{[l]}$ is the activation function at layer l, L is the number of layers in the network, and $A^{[l]}$ is the predicted output vector. The values of the weight and bias vectors are updated via backpropagation. These equations apply to the general category of ANNs, but it should be noted that specific DNN varieties such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have shown promise in language modeling and computer vision applications, respectively.

B. LINEAR REGRESSION

Linear Regression is a special case of the General Linear Model (GLM), a method to determine if a linear relationship exists between two or more variables which can be expressed as an equation with linear coefficients. Simple linear regression is a special case of the GLM which has a fixed intercept, and fixed linear coefficient. GLM and Linear Regression are often used in Machine Learning research to model relationships between multiple variables [11].

C. ANOVA

Analysis of Variance (ANOVA) is a special case of GLM which is used to determine whether two variables are from the same distribution and has a set of assumptions which include normality of the data distribution and homogeneity of variances. However, some methods have less assumptions for non-normally distributed data such as Kruskal-Wallis One-Way ANOVA [12], which is used in this research.

IV. MATERIALS AND METHOD

This research proposed a novel approach, Small Set of Linearized Variables (SSOLV) to network traffic flow analysis as described in this section.

Before performing feature selection two openly available datasets were selected for analysis. The details of the datasets and analysis are in the remainder of this section.

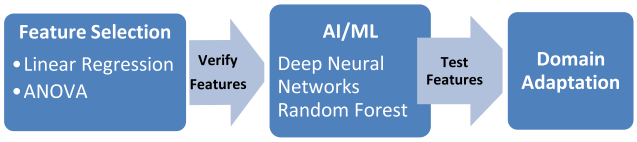

FIGURE 2. SSOLV process diagram.

A. DATASETS

Openly available datasets were selected from literature to enable re-producibility of results. The popularly cited UNSW-NB15 dataset was selected from literature for initial analysis and to determine a common set of features available from the Zeek network monitoring tool, which was used to generate the dataset. The Training and Testing datasets were selected since they are curated by the dataset authors to have a similar distribution to the overall data collected and are intended for AI/ML applications [4]. To verify the features, the popular KDDCUP99 dataset was selected, and the 10 percent subset was used for testing [13]. A key advantage of the UNSW-NB15 dataset over the KDDCUP99 dataset is in the removal of duplicate records, and incorporation of modern malware attacks [4]. These measures improve the overall statistical conclusion validity of the results since statistics often assumes that the statistics are similar to the overall population of the observed phenomena [9]. In addition, for incorporation of more modern cyberattacks data, the CICIDS2017 dataset was selected as a second verification of the features and has the advantage that it is a more recent dataset than the KDDCUP99, and UNSW-NB15 datasets [5].

All datasets are in the open CSV format to ensure reproducibility. The datasets were also selected on a basis of their span in years and coverage of different computing environments. Since the UNSW-NB15 dataset and KDDCUP99 dataset are 16 years apart [8], result that perform well in both datasets are likely to extend to different dataset. In addition, since the UNSW-NB15 and CICIDS2017 datasets are 2 years apart, results that perform well in both datasets are likely to extend to different datasets and more modern datasets.

B. ANALYSIS

Initial analysis of the data indicated that source bytes (sbytes), destination bytes (dbytes), source packets (spkts), and destination packets (dpkts) were real time features available from the Zeek network monitoring tool that are available in all datasets. Linear Regression was used to determine the independence of relevant features and then ANOVA was used to verify the uniqueness of the data. The statistical properties of the data were then examined, and a transformation was performed on the data to improve the normality of the data distribution. The transformation, which uses the natural logarithm is also a method used to perform linearization of variables. The data statistical properties were then examined before and after the transformation to verify that the statistical properties were normalized after transformation.

Four additional features were then derived from the transformed features and all features were tested with a deep learning neural network and random forest on all datasets to determine accuracy, precision, and recall performance metrics. The assumption is that the AI/ML algorithms would perform better on data that follows a more normal distribution since a normal distribution is an assumption of many statistical and regression algorithms [9].

C. ENVIRONMENTS

This research used several environments and software suites to perform data preparation and cleansing, descriptive and inferential analysis, and AI/ML modeling and metric gathering. Prior to analysis, the dataset CSVs were loaded and cleansed using Microsoft Excel. Afterwards. statistical analysis was performed in RStudio version 2022 and R version 14, to perform descriptive and inferential analysis on the cleansed dataset CSVs. Lastly, AI/ML analysis and metric gathering was performed in the Google Colab®Pro cloud-based environment.

V. EXPERIMENT

RStudio version 2022 and R version 14 were used for statistical analysis and Google Colab®Pro was used for AI/ML analysis and metric gathering.

The UNSW-NB15, KDDCUP99 and CICIDS2017 CSVs were cleaned so that column heading appeared at the top of the CSVs. Then features were derived from the UNSW-NB15 CSV file using RStudio. Prior to analysis, linear regression was used to determine independent features which were verified using ANOVA.

Afterwards, the features were transformed in Google Colab Pro by loading the dataset CSVs. The TensorFlow and Keras libraries were used to create a multilayer deep neural network, and the Scikit Python library was used to implement random forest. Then the Scikit **classification_report** function was used to retrieve the accuracy, precision, recall and macro average metrics of the AI/ML algorithm performance.

Lastly, to demonstrate transferability of results, the ADAPT Python Library was used to implement domain adaptation from the UNSW-NB15 dataset to the KDDCUP99 and CICIDS2017 datasets and test the adapted dataset on the KDDCUP99 and CICIDS2017 datasets [14].

The UNSW-NB15 dataset was used to derive the initial features, and transformations which include Linear Regression, ANOVA within RStudio. The remaining portion of the experiment then used Google Colab®Pro and Python for implementation. The final transformed features were trained on the UNSW-NB15 training set and tested on the UNSW-NB15 dataset on both the Random Forest and Deep Learning network. Afterwards the transformed features were tested using a randomly selected without replacement subset of 80% of the KDDCUP99 10 Percent dataset and tested on the remaining 20% on both the Random Forest and Deep Learning network. Lastly, the transformed features were tested using a randomly selected without replacement subset of 80% of the CICIDS2017 dataset and tested on the remaining 20% on both the Random Forest and Deep Learning network.

After performance metrics were gathered for each dataset, the ADAPT Instance-Based Feature Augmentation (FA) domain adaptation algorithm [14] was used to adapt the UNSW-NB15 dataset using only the first 100 data points of the KDDCUP99 10 percent dataset and CICIDS2017 datasets and tested on the full KDDCU99 10 percent and CICIDS2017 datasets respectively for both the Random Forest and Deep Learning network.

VI. RESULT

This section contains the results of the statistical analysis, and AI/ML experimental results. The first two subsections contain the results of statistical analyses which include linear regression and ANOVA. Next, the third subsection contains feature transformation results. The last subsection then includes the results of AI/ML modeling which include Random Forest, and a Deep Neural Network and details on the results of domain adaptation.

A. LINEAR REGRESSION

The simple linear regression revealed that source packets and source bytes have a strong linear relationship with a low p-value ($p<0.001$), and that destination packets and destination bytes have a strong linear relationship with ($p<0.001$). In addition, linear regression on source bytes and destination bytes suggest that these features are independent ($p<0.001$). Therefore, source bytes and destination bytes were selected as initial independent features. Figs. 3, 4, and 5 show the results on linear regression.

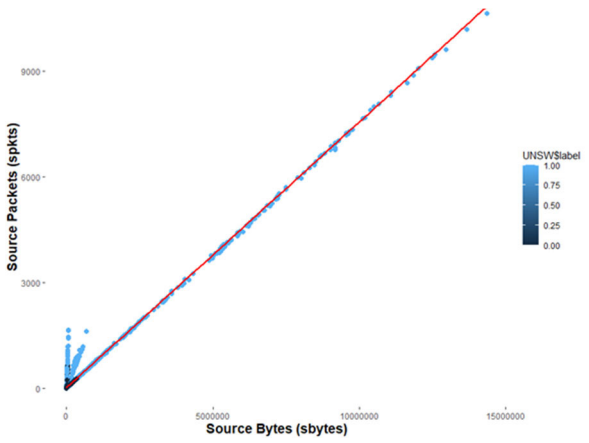

FIGURE 3. Source packets vs. source bytes linear regression.

B. ANOVA

Afterwards, statistical analysis was performed on the source bytes and destination bytes features showing that they are right-skewed and leptokurtic. Therefore, methods tolerant to non-normal distributions of data such as Kruskal-Wallis one-way ANOVA should be used for further analysis [9], since due to the severe non-normal distribution of the data, the

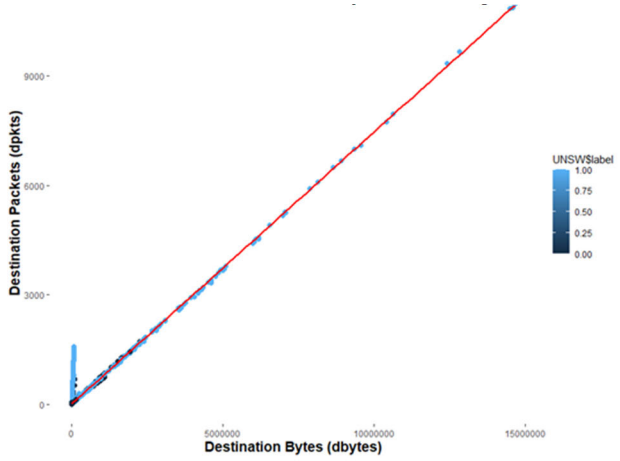

FIGURE 4. Destination packets vs. destination bytes linear regression.

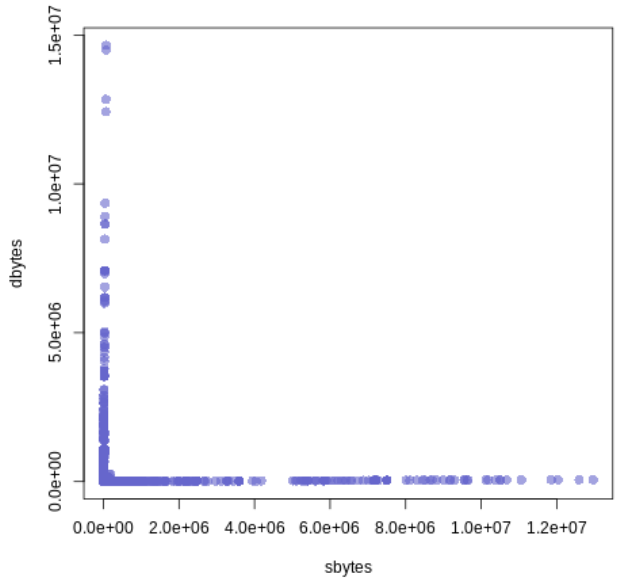

FIGURE 5. Destination bytes vs. source bytes linear regression.

standard factorial ANOVA is not expected to yield statistically significant results. Kruskal-Wallis one-way ANOVA with a sample of 10,000 data points was then used to verify that source bytes and destination bytes and the result is that source bytes and destination bytes are independent with a significance of$p<0.001$ indicating that the results are statistically significant, since$p$is less than 0.05. Prior to conducting Kruskal-Wallis ANOVA, the assumption of equal variances was tested using Levene's test, which indicated with a high degree of confidence ($p<0.001$) that the variances between source bytes and destination bytes were significantly different for Source Bytes [15], [16].

Before the null hypothesis can be rejected in the ANOVA analysis, Type 1 error should be controlled by one or more post-hoc analyses, which each have their pros and cons. The Bonferroni test was selected due to it conservativeness, and tolerance to non-normal distributions of data [16].

Since the adjusted P value as a result of the Bonferroni Test is still very low ($p<0.001$), the null hypothesis that source bytes and destination bytes are from the same distribution can be rejected with a high degree of confidence due to the low probability of Type I error [16].

Table 1 shows the results of statistical analysis on source bytes and destination bytes before transformation and Table 2 shows the results of Kruskal-Wallis one-way ANOVA.

TABLE 1. Statistical analysis before transformation.

Variable	M	Sd	Skewness	Kurtosis
Sbytes	8,573	173,773.9	47.91689	2,616.499
Dbytes	14,387	146,199.3	44.33998	3,272.347

TABLE 2. Kruskal-Wallis One-Way ANOVA of source bytes and destination bytes before transformation.

Chi-squared	Df	P
186,247	8,652	<0.001

Source Bytes and Destination Bytes were transformed by taking their natural logarithm and statistical analysis was re-performed. The results indicate that the transformation normalized source bytes and destination bytes. Table 3 shows the statistics of the transformed source bytes and destination bytes and Table 4 shows the results of Kruskal-Wallis one-way ANOVA of the transformed source bytes and destination bytes using 10,000 randomly selected data points.

TABLE 3. Statistical analysis of transformed source bytes and destination bytes after transformation.

Variable	M	Sd	Skewness	Kurtosis
Sbytes	6.292	1.645736	1.1464537	1.606885
Dbytes	−6.867	15.19052	−0.103887	−1.947333

TABLE 4. Kruskal-Wallis One-Way ANOVA of source bytes and destination bytes after transformation.

Chi-squared	Df	P
186,247	8,652	<0.001

C. FEATURE TRANSFORMATION

Afterwards, since source bytes and destination bytes are independent, four additional features were derived from the transformed features; the logarithm of source bytes plus the logarithm of destination bytes, logarithm of source bytes minus the logarithm of destination bytes, logarithm of source bytes multiplied by the logarithm of destination bytes, logarithm of source bytes divided by the logarithm of destination bytes. These features were added to the two transformed

features logarithm of source bytes and the logarithm of destination bytes, for a total of six features. In the transformed features, logarithm is the natural logarithm and a small 10^{-10} value is added to source bytes and destination bytes before calculating the logarithm to prevent a division-by-zero error.

D. AI/ML AND DOMAIN ADAPTATION

These six features were tested on both the UNSW-NB15 and KDDCUP99 datasets for both the Random Forest and Deep Learning networks. The classification metrics were gathered via the Scikit *classification_report* feature. Tables 5 6 and 7 show the macro average performance metrics for the UNSW-NB15, KDDCUP99 and CICIDS2017 datasets for both the Random Forest and Deep Learning networks.

TABLE 5. Performance metrics for UNSW-NB15.

Network	Precision	Recall	F1-score
Random Forest	87%	83%	84%
Deep Neural Network	87%	86%	86%

TABLE 6. Performance metrics for KDDCUP99.

Network	Precision	Recall	F1-score
Random Forest	99%	97%	98%
Deep Neural Network	98%	96%	97%

TABLE 7. Performance metrics for CICIDS2017.

Network	Precision	Recall	F1-score
Random Forest	98%	92%	94%
Deep Neural Network	96%	90%	93%

Afterwards, the ADAPT Instance-Based Feature Augmentation (FA) domain adaptation algorithm [14] was used to adapt the UNSW-NB15 dataset using only the first 100 data points of the KDDCUP99 10-percent dataset and tested on the full KDDCU99 10-percent dataset for both the Random Forest and Deep Learning network. Tables 8 and 9 show the results of domain adaptation between the UNSW-NB15 and KDDCUP99 and CICIDS2017 datasets respectively.

TABLE 8. Domain adaptation results for the UNSW-NB15 and KDDCUP99 Datasets.

Network	Precision	Recall	F1-score
Random Forest	97%	94%	96%
Deep Neural Network	97%	94%	96%

TABLE 9. Domain adaptation results for the UNSW-NB15 and CICIDS2017 datasets.

Network	Precision	Recall	F1-score
Random Forest	89%	80%	83%
Deep Neural Network	89%	80%	83%

The results for the KDDCUP99 dataset and the domain adaptation between UNSW and KDDCUP99 datasets have high accuracy (>95%) with a high precision (>95%) and recall (>90%) rate. Likewise, the results for the CICIDS2017 dataset and the Random Forest domain adaptation datasets have high accuracy (>95%) with a high precision (>95%) and recall (>90%) rate with exception of the Deep Learning domain adaptation which had medium accuracy (89%) and recall (80%). Therefore, the features are both relevant and able to yield high accuracy results across different datasets especially when using machine learning methods such as Random Forest. The domain adaptation differences for the UNSW-NB15 dataset to CICIDS2017 dataset may be attributable to the fact that the CICIDS2017 dataset includes modern cyberattacks not captured in the KDDCUP99 and UNSW-NB15 datasets [5].

The improvement of performance due to the transformation are evident in the Confusion Matrix diagrams of the Random Forest and Deep Neural Networks on the transformed features. Fig. 6 shows the transformed features performance for the Random Forest Machine learning algorithm on the UNSW-NB15 dataset.

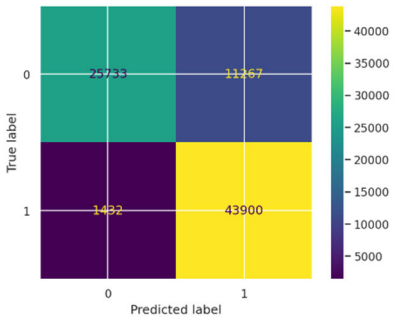

FIGURE 6. UNSW-NB15 random forest confusion matrix.

Fig. 6 shows a low false positive rate and overall high accuracy. Fig. 7 shows the transformed features performance for the Deep Learning algorithm on the UNSW-NB15 dataset.

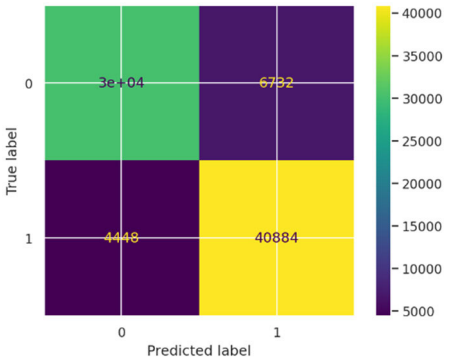

FIGURE 7. UNSW-NB15 deep learning confusion matrix.

The performance was slightly lower than random forest in precision and false positive rate. Fig. 8 shows the transformed

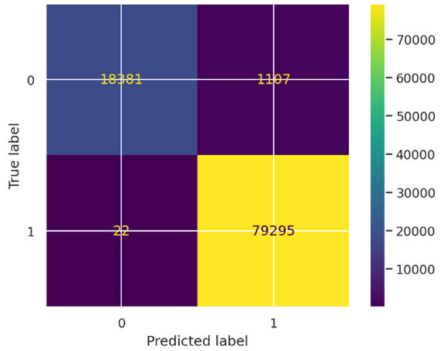

FIGURE 8. KDDCUP99 random forest confusion matrix.

features performance for the Random Forest Machine Learning algorithm on the KDDCUP99 dataset.

Fig. 8 shows high performance in terms of true positives with a very low false positive rate. Fig. 9 shows the transformed features performance for the Deep Learning algorithm on the KDDCUP99 dataset.

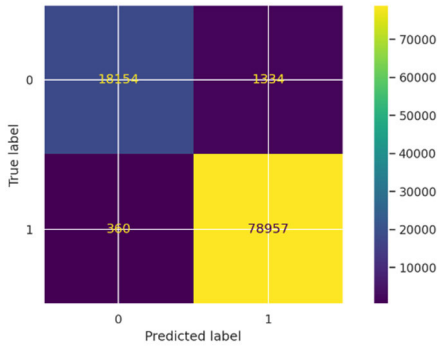

FIGURE 9. KDDCUP99 deep learning confusion matrix.

Fig. 9 shows high performance in terms of true positives with a very low false positive rate. Fig. 10 shows the transformed features performance for the Random Forest Machine Learning algorithm on the CICIDS2017 dataset.

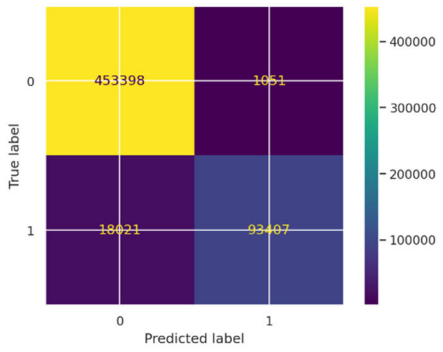

FIGURE 10. CICIDS2017 random forest confusion matrix.

Fig. 10 shows high performance in terms of low false positive rate and overall high accuracy. Fig. 11 shows the transformed features performance for the Deep Learning algorithm on the CICIDS2017 dataset.

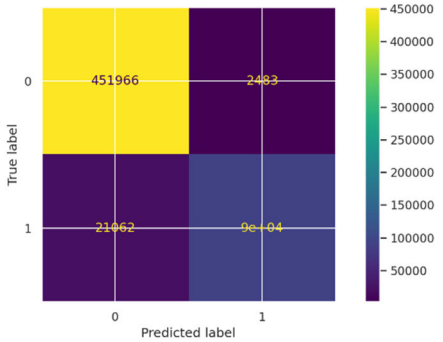

FIGURE 11. CICIDS2017 deep learning confusion matrix.

Fig. 11 shows high performance in terms of low false positive rate, and overall high accuracy. Fig. 12 shows the transformed features performance for the Machine Learning algorithm on Domain Adaptation from the UNSW dataset to the KDDCUP99 dataset using the FA algorithm.

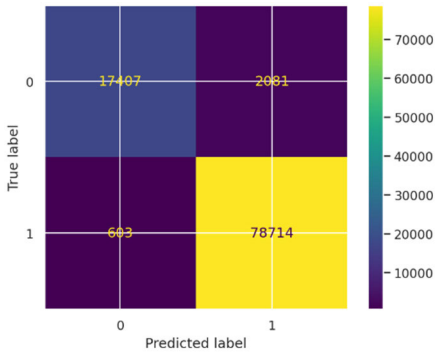

FIGURE 12. UNSW-NB15 to KDDCUP99 domain adaptation random forest confusion matrix.

Fig. 12 shows high performance in terms of low false positive rate, but a lower performance on recognizing true negatives. Fig. 13 shows the transformed features performance

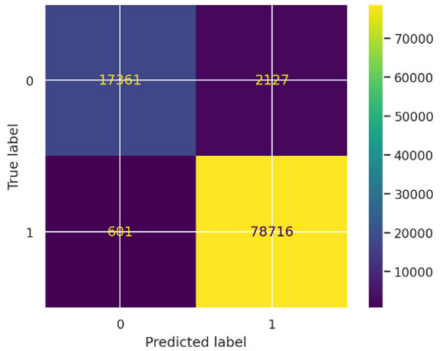

FIGURE 13. UNSW-NB15 to KDDCUP99 domain adaptation deep learning confusion matrix.

for the Deep Learning algorithm on Domain Adaptation from the UNSW dataset to the KDDCUP99 dataset using the FA algorithm.

Fig. 13 shows high performance in terms of low false positive rate, but a lower performance on recognizing true negatives. Fig. 14 shows the transformed features performance for the Machine Learning algorithm on Domain Adaptation from the UNSW dataset to the CICIDS2017 dataset using the FA algorithm.

Fig. 14 shows high performance in terms of low false positive rate, but low accuracy on recognizing true positives. Fig. 15 shows the transformed features performance for the Deep Learning algorithm on Domain Adaptation from the UNSW dataset to the CICIDS2017 dataset using the FA algorithm.

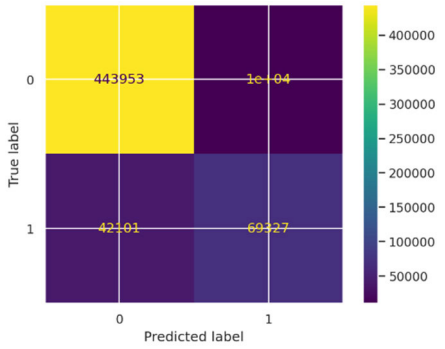

FIGURE 14. UNSW-NB15 to CICIDS2017 domain adaptation random forest confusion matrix.

Fig. 15 shows high performance in terms of low false positive rate, but low accuracy on recognizing true positives. These findings suggest that a different domain adaptation method may be necessary to obtain better performance metrics on some domains.

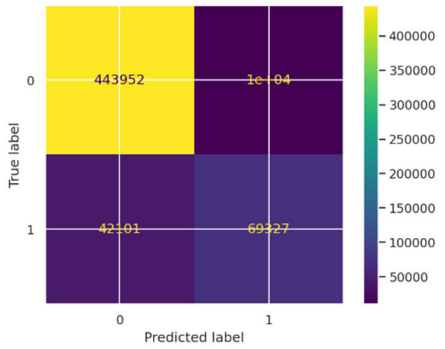

FIGURE 15. UNSW-NB15 to CICIDS2017 domain adaptation deep learning confusion matrix.

DISCUSSION

Source Bytes and Destination Bytes were determined to be independent features via statistical analysis which used methods tolerant to non-normal distributions of data such as Kruskal-Wallis one-way ANOVA. Most cases have high performance with high accuracy (>95%) and high precision (>95%) and recall (>90%) rate with exception of the Random Forest UNSW-NB15 results. Including domain adaptation cases demonstrate that the six derived and transformed features are applicable to network traffic from different systems as demonstrated in tests using the popular KDDCUP99 dataset, and domain adaptation between the UNSW-NB15 and KDDCUP99 and CICIDS2017 datasets.

The six features are the logarithm of source bytes, the logarithm of destination bytes, logarithm of source bytes plus the logarithm of destination bytes, logarithm of source bytes minus the logarithm of destination bytes, logarithm of source bytes multiplied by the logarithm of destination bytes, logarithm of source bytes divided by the logarithm of destination bytes where logarithm is the natural logarithm and a small 10^{-10} value is added to source bytes and destination bytes before calculating the logarithm to prevent a division-by-zero error.

CONCLUSION AND FUTURE WORK

In this work, we proposed an approach for identifying transformed features via statistical analysis, which can be used in AI/ML-based systems. We also present a deep learning framework for training neural networks based on labeled Zeek datasets containing both malicious and benign activity in real or near-real time. In addition, we presented a mechanism for transfer learning using domain adaptation techniques to adapt neural networks trained on one labeled Zeek dataset to another neural network trained on a different labeled Zeek dataset. Our results show that we can classify malicious activity with up to 97-99% accuracy in select cases and high precision (>95%) and recall (>90%) rate. This framework demonstrates a mechanism that stand-alone systems disconnected from larger networks can use to recognize adversarial activity in real time and is transferrable to other stand-alone systems.

The six features identified are the logarithm of source bytes, the logarithm of destination bytes, logarithm of source bytes plus the logarithm of destination bytes, logarithm of source bytes minus the logarithm of destination bytes, logarithm of source bytes multiplied by the logarithm of destination bytes, logarithm of source bytes divided by the logarithm of destination bytes where logarithm is the natural logarithm and a small 10-10 value is added to source bytes and destination bytes before calculating the logarithm to prevent a division-by-zero error. Most cases have high performance with high accuracy (>95%) and high precision (>95%) and recall (>90%) rate with exception of the Random Forest UNSW-NB15 results. Including a domain adaptation case demonstrates that the six derived and transformed features are applicable to network traffic from different systems as demonstrated in tests using the popular UNSW-NB15, KDDCUP99 and CICIDS2017 datasets.

It is recommended that additional datasets be tested using these six features to determine if the results hold for future

research. To apply these results to current Zeek traffic for a particular network, it is recommended that a small, labeled dataset ($N = 100$) be obtained from the target domain so that domain adaptation can be used to adapt a trained network on a public dataset such as UNSW-NB15 to the target domain as done in this research in the cases of UNSW-NB15 to KDDCUP99 and UNSW-NB15 to CICIDS2017.

In addition, it is recommended that additional research be done to identify additional features which can be transformed in real or near real time. The inclusion of additional unique features can result in accuracy and precision and recall performance improvements on the UNSW-NB15 and CICIDS2017 datasets.

ACKNOWLEDGMENT

The authors would like to thank the following personnel for assistance in acquiring datasets: Gary Huntress for the DNS logs and Zeek network sensor datasets; Rabbil Ahmed and Dale Walters for PCAP data. They would also like to thank Jim Kasischke for help in filing the patent application and Tony Ruffa, Mark Dalton, Rebecca Chhim, and Paul Boivin for support of artificial intelligence, machine learning, and cybersecurity science and technology projects.

REFERENCES

[1] V. Malhotra, M. S. Watchorn, K. L. Sharkey, D. Chanda, A. H. Carlson, M. Lizar, P. Gupta, M. A. Enright, A. Polyakov, and D. Reynolds, *Cybersecurity for Next-Generation Connectivity Systems-Rethinking Digital Architectures to Safeguard the Next Generation From Cybersecurity Breaches*, IEEE, 2022.

[2] W. Wu, L. He, W. Lin, Y. Su, Y. Cui, C. Maple, and S. Jarvis, "Developing an unsupervised real-time anomaly detection scheme for time series with multi-seasonality," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 9, pp. 4147–4160, Sep. 2022.

[3] N. Moustafa and J. Slay, "The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems," in *Proc. 4th Int. Workshop Building Anal. Datasets Gathering Exper. Returns Secur. (BADGERS)*, Kyoto, Japan, Nov. 2015, pp. 25–31.

[4] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Canberra, ACT, Australia, Nov. 2015, pp. 1–6.

[5] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021.

[6] B. Drozdenko and M. Powell, "Utilizing deep learning techniques to detect zero day exploits in network traffic flows," in *Proc. IEEE 13th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2022, pp. 0163–0172.

[7] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, *Zero Trust Architecture*, document Special Publication (NIST SP) 800, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, 2020.

[8] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset," *Inf. Secur. J., Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, 2016.

[9] V. Trajkovski, "How to select appropriate statistical tests in scientific articles," *J. Special Educ. Rehabil.*, vol. 17, nos. 3–8, pp. 5–28, 2016.

[10] A. Salami, F. Ghassemi, and M. H. Moradi, "A criterion to evaluate feature vectors based on ANOVA statistical analysis," in *Proc. 24th Nat. 2nd Int. Iranian Conf. Biomed. Eng. (ICBME)*, Tehran, Iran, Nov. 2017, pp. 14–15.

[11] J. M. Górriz, C. Jiménez-Mesa, F. Segovia, J. Ramírez, S. Group, and J. Suckling, "A connection between pattern classification by machine learning and statistical inference with the general linear model," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 11, pp. 5332–5343, Nov. 2022.

[12] T. V. Hecke, "Power study of Anova versus Kruskal–Wallis test," *J. Statist. Manage. Syst.*, vol. 15, nos. 2–3, pp. 241–247, May 2012.

[13] S. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. Chan. (Dec. 13, 1998). *KDD Cup 1999*. Univ. California Irvine (UCI). Accessed: Sep. 5, 2023. [Online]. Available: https://archive.ics.uci.edu/dataset/130/kdd+cup+1999+data

[14] A. de Mathelin, M. Atiq, G. Richard, A. de la Concha, M. Yachouti, F. Deheeger, M. Mougeot, and N. Vayatis, "ADAPT: Awesome domain adaptation Python toolbox," 2021, *arXiv:2107.03049*.

[15] G. D. Garson, *Testing Statistical Assumptions*. Asheboro, NC, USA: Statistical Publishing Associates, 2012.

[16] S. Lee and D. K. Lee, "What is the proper way to apply the multiple comparison test?" *Korean J. Anesthesiol.*, vol. 71, no. 5, pp. 353–360, Oct. 2018.

MAKIA S. POWELL (Member, IEEE) received the B.S. degree in electrical and computer engineering and the M.S. degree in computer engineering specializing in solid-state devices from the University of Massachusetts Dartmouth, in 2007 and 2010, respectively. She is currently pursuing the Ph.D. degree in data science from National University. In September 2008, she joined NUWC after fulfilling degree requirements for her master's degree. She is also a Computer Engineer at the Naval Undersea Warfare Center Division Newport (NUWCDIVNPT). She has authored multiple patents in her field of study including U.S. Patent 8 725 786 "Approximate SRT Division Method," US9417839B1 "Floating Point Multiply-Add-Subtract Implementation," U.S. Patent 11 620 106 "Parallel Hybrid Adder." She was a principal investigator for FY14 and FY15 Workforce development projects whose goal was to yield optimized hardware on field programmable gate arrays (FPGAs). The results of the project applied to solid-state devices in general which include microprocessors. The project also resulted in a patent filed through NUWC and the refinement of a U.S. Patent 11 620 106. Since 2019, She has been involved in machine learning projects and has experience in analyzing, clustering, and categorizing raw PCAP data from publicly available datasets using Python and TensorFlow.

BENJAMIN M. DROZDENKO (Member, IEEE) received the B.S. degree in computer science and computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in May 2004, and the M.S. degree in electrical engineering with a concentration in communications and signal processing and the Ph.D. degree in computer engineering from Northeastern University, Boston, MA, USA, in May 2007 and May 2017, respectively. His dissertation was titled, "Enabling Protocol Coexistence: Hardware-Software Co-design of Wireless Transceivers on Heterogeneous Computing Architectures." From 2008 to 2014, he was with MathWorks as a Training Engineer and Signal Processing Content Specialist. From September 2017 to August 2020, he was an Assistant Professor of cyber engineering and computer science with Louisiana Tech University, Ruston, LA, USA. He is currently the Cybersecurity Science and Technology (S&T) Lead at the Naval Undersea Warfare Center Division Newport (NUWCDIVNPT). He is also an established researcher from academia and with ten years of experience as an engineer in industry. He has published in various technical journals, such as IEEE Transactions on Emerging Topics in Computing and IEEE Access, as well as conferences, including FPL, IEEE GreenTech, IEEE GLOBECOM, IEEE DCOSS, IEEE INFOCOM workshops, and CROWNCOM.