Project Title

Improving Network Intrusion Detection Using "KISTI+IDS2021-CDMC " Dataset, machine learning, and Deep learning techniques.

Submitted by

Shrief Ahmed, Amjad Dife, Mark Messih, and Amir Youssef.

Institution

University of Ottawa

Course

Artificial Intelligence for Cybersecurity Applications

Date

Fall 2022

1. problem statement

Network intrusion detection is defined as a pattern recognition problem where the events that constitute a normal network flow pattern are recognized, and an intrusion is defined as events that don't fit an expected pattern allowing the categorizing of events into binary (normal or abnormal network flow) or multiple classes considering the pattern of each type of attack.

2. Outline of Existing Works

date	Title of Work	Author Name
Date of Conference: 11-14 December 2017	"Network intrusion detection using word embeddings,"	▪ Xiaoyan Zhuo ▪ Jialing Zhang ▪ Seung Woo Son
Published: 05 Jul 2018	"TR-IDS: Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest"	▪ Erxue Min ▪ Jun Long ▪ Qiang Liu ▪ Jianjing Cui ▪ Wei Chen
First Online: 16 September 2018	“WEDL-NIDS: Improving Network Intrusion Detection Using Word Embedding-Based Deep Learning Method.”	▪ Jianjing Cui ▪ Jun Long ▪ Erxue Min ▪ Yugang Mao
Published: 24 June 2021	“Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges”	▪ Geeta Kocher ▪ Gulshan Kumar

3. Problem description

A network intrusion detection system is tasked with the crucial task of finding abnormal patterns of network flow that can help the system protect the network from different types of attacks, so an accurate depiction of this problem would be to find features that can accurately represent the normal network behavior, but with the everyday increase in the scale of data shared through networks, it becomes difficult to differentiate between normal and abnormal network traffic, and thus the need for machine learning and deep learning techniques because such techniques are known for their ability to handle large amounts of data, and extract relevant features that can help the system classify traffic passing through the network. The main source of features can be collected through network packets which have some statistical features such as the source and destination of the packets as well as packet payload, the payload is often present as bytes which when used in raw format would be difficult for Ai techniques to infer helpful features from, and thus NLP techniques are essential in converting information present in the payload into helpful features for the AI-based intrusion detection system to use. In conclusion, the problem scope as stated above is building an effective network intrusion detection system by using advanced machine and deep learning algorithms.

4. Comparative Review

In [1] the authors tried to fill the technological gap faced by conventional data analysis and knowledge extraction processes deployed in various cybersecurity-related problems using word embedding models, this largely reduces data processing work, e.g., data cleaning, normalization, correlation analysis, or matrix manipulation.

They used dataset KDD CUP' 99 in their research and evaluated the performance of their model by varying vectorizers and classifiers. **First**, they labeled the data into binary and multi-class classes, where the data contains 37 attack types and normal type, which can be categorized into DOS, Probe, U2R, R2L, and Normal, while Each log comprises 41 features, which can be divided into basic features, content features, and traffic features. **Second**, they divided the network logs in the data into training and testing data with different proportions to explore the impact of data size on the prediction performance. **Third**, they used three vectorizers which are Count Vectorizer, TFIDF Vectorizer, and word2Vec to obtain representation vectors during the training. Pre-trained word vectors were obtained from the word2vec toolkit by setting the Skip-gram model vector size to 300 and windows size to 8. **Fourth**, they used Naive Bayes, Logistic Regression, and SVM for the classification and explored the impact of data size on the prediction performance by varying the size of training data. They used SGD Classifier in Scikit-learn and applied Multinomial NB and Gaussian NB in Scikit-learn for Naïve Bayes. **Fifth**, they used error metrics of precision, *Recall*, *a*nd *F*1 Score to evaluate the performance. Sixth, they used PCA to visualize high dimensional vector representation of word embeddings and they visualized precision, *Recall* and *F*1 Score to show and compare the performance between different models easily.

In [2] the authors define the job of Intrusion detection systems (IDS) as detecting attacks in the network flow, two techniques could be employed for such detection to happen, one relies on the rules developed by cyber security experts for each protocol such as HTTP, FTP, and SMTP, these are called manual parsers which prove to be good at recognizing the known attacks but incapable of detecting new attack patterns, the second technique relies on deep learning techniques that aim to model what the normal network flow would look like and then define the attack as an anomaly compared to the normal behavior. The paper focuses on implementing a novel architecture for an IDS that relies on the second technique mentioned above by using statistical features collected from IDS alert packets as well as word embeddings of the payload present in the packets to achieve superior accuracy on the ISCX2012 dataset.

They developed an IDS that uses both statistical features and payload word embeddings to perform multiclass classification of network flow into legitimate or malicious while considering various types of attacks.

The methods are built on considering each byte to be a word and embedding it into a low dimensional vector for lowering computational costs, then combining the bytes of each packet together to form a sentence and apply padding to ensure that all the sentences have the same length, then pass each sentence representation to a slightly modified version of Text CNN where a feature layer is added before the SoftMax classification layer in order to extract the feature representation of CNN and use it along with the statistical features as an input for random forest classifier, all while taking care to produce a near balanced sample of the data to be able to use evaluation metrics like accuracy.

In [3] the main goal of this research is to build an improved IDS by using deep learning and word embedding. and this NIDS performs dimensionality reduction and learns features from the data so it can detect intrusion more effectively than the other methods.

The authors were clear when they defined the challenges behind building network intrusion detection systems. They defined two challenges: 1. feature engineering and dimension reduction. They handled the first challenge by using deep learning techniques and handled the second challenge by using word embedding. So, by using deep learning they didn't have to perform feature engineering, and by using word embedding they can reduce the dimensionality without wasting the similarity relationship.

They have mentioned the usage of deep learning methods in intrusion detection. They divided the contributions of deep learning into three parts:

1. deep learning as classifiers: in this case, we have the row data and use deep learning to train a classifier or even to select a feature and use traditional machine learning methods to train the classifiers. They defined that the weakness of this point is the ignorance of using deep learning to learn from the row data itself.

2. deep learning as a feature extractor and they mentioned the "Auto-Encoder" as an example.

3. deep learning as an end-to-end model. They defined the usage of deep learning as a combination of the two previous parts, which can be used first to be applied to the row data and then train the classifiers. They mentioned CNN and RNN as two approaches that have been used within this part. They defined the limitations as the process of transferring the traffic data into images before the experiment. They mentioned that the method they implemented was the same as the third approach with some improvements.

As they used supervised machine learning techniques, they were expecting that the model would not perform well with unknown data. they refer to the method they implemented by "WEDL-NIDS" which stands for word embedding and deep learning-based network intrusion detection system. in WEDL-NIDS they use word embedding as a first step to reduce the dimension of the payload of the packet, then as a second step they use CNN to learn the features of the network traffic. Finally, they use LSTM.

They defined their model as 3 stages model: data preprocessing, dimensionality reduction, and deep learning architecture. in the data preprocessing phase, the raw network traffic has been transformed into head features and payload text, then the payload text was input into the phase of dimensionality reduction in which the output is word vectors. Finally, the output of the two previous phases was input into the deep learning model. in the data preprocessing phase, they extract some features that identify the packets like length, window_size, option_num, and other features. in the phase of extraction of the payload, they decided to choose only the first 100 words from each packet's payload, and they handled the problem of the length by truncating the extra if it was more than 100 words, and by padding zeros if it was less than 100. Finally, they used a one-hot encoding technique to transform each word in the text into a 256-dimension vector, they did that as a preparation step for the next phase. in the dimensionality reduction phase, they mentioned that they had to choose between Word2Vec and GloVe techniques. And they chose the Word2Vec technique and skip-Gram model specifically. as an output of this step, they got the word vectors. They divided the deep learning phase into two main steps learning payload features in which they used CNN and learning global features in which they used LSTM and RNN.

In [4] A lot of researchers have used **SVM** for detecting intrusions. To get a high detecting efficiency a high training data quality is required. The authors have introduced a logarithm marginal density ratio transformation (LMDRT) to get a better-quality detection. There have been 4 datasets chosen to be tested on this Algorithm which are: "**UNSW-NB15", "NSL-KDD", "Kyoto 2006" and "CICIDS2017"**

A new method has been developed with a combination of two classifiers **Random tree** and **Naïve Bayes** tree. The algorithm has been tested on the "**NSL-KDD"** dataset. Another hybrid layered intrusion detection systems method was implemented on various machine learning methods and was trained and tested on the "**NSL-KDD"** dataset. Normalization and transformation operations were performed.

Another hybrid method had been developed to achieve a high DR with a Low false positive rate; the method was implemented in two stages. In the first stage to construct the detection portion a low-complexity anomaly detection method was built, in the second step, **KNN** Algorithm was used. The two stages' purpose is to reduce the false positive and false negative numbers.

The **logistic regression** approach core is for nonlinear fixed Attributes of internet traffic. This approach separates different types of Dos attacks in complex network flows, achieving high results in terms of precision and accuracy.

Based on the **Random Forest** classifier a new proposed model was used as an ensemble classifier. The model has been tested on the "**NSL-KDD"** dataset and the "**KDD cup"** dataset. And it ended up outperforming all the other models in terms of attack classification.

5. Results in Available Research

As a result of the experiment in [1]

For binary classification (normal vs. abnormal), When the training data size is 10%, 50%, or 80% out of the entire dataset, the F1 score of classification is in the range between 0.90 and 0.97, and the F1 score is above 0.72 even when the training data size is only 0.1% of the total dataset.

After a set of comparisons, they concluded that the Word2Vec, Count Vectorizer, and TFIDF Vectorizer can extract features well when data is large enough, furthermore the features extracted by the Word2Vec vectorizer are more robust even when the training data size is limited.

Similarly, Naive Bayes, Logistic Regression, and SVM all perform well in classification tasks when training data is sufficient. Generally, SVM performs slightly better than Logistic Regression and Naive Bayes regardless of which vectorizer is used, and the classification performance obtained from Naive Bayes drops more than that of the other two algorithms as the training data size gets smaller. Therefore, they evaluated the run time of classification algorithms where the three used took less than one minute on 311,029 data samples, which meets the efficiency requirement in network intrusion detection while they found that the run time increased to over 2 hours on the same device when CNN model was run on the same device but it achieved a higher accuracy of 0.99, so this meant that there is a trade-off between the classification performance and the runtime.

For classifying network log instances into five classes (DOS, Probe, U2R, R2L, Normal), Similarly, when the training data is 10%, 50%, or 80% of the total dataset, the F1 score of classifiers ranges from 0.88 to 0.97. SVM performed slightly better than Logistic Regression and Naive Bayes. Word2Vec vectorizer maintained stable high performance even with a smaller training data size. These results demonstrated that their NLP-based models could achieve high performance on the multi-class classification of non-textual network log data.

As a result of the experiment in [2]

The authors claimed that they are the first to use a combination of the statistical features of the packets and the payload embedded in it, it was also claimed that random forest would be a good fit for such a classification problem.

Those claims were well supported by results as the random forest was able to achieve an accuracy of 99 percent, and several experiments were performed by using different features such as raw bytes along with the statistical features, as well as experimenting with using the statistical features only, such experiments were able to prove that the CNN feature representation performs better than using raw bytes or not using payload information.

As a result of the experiment in [3]

They claimed that the features' design has a high effect on the performance of the intrusion detection systems. in the second claim, they mentioned that the features' design takes a lot of time. They mentioned that their approach achieves 99.97% in terms of accuracy, and 96.39% in terms of detection rate mentioned which dropped to 90.00% when considering the Infiltration attack. and the 0.94 F1-score and they considered that as a quite good performance. Finally, unlike the way they used to show the result in terms of accuracy, detection rate, and F1-score, the authors used unclear language to show the result of the model in terms of false alarm rate, they said "The FAR was pretty good".

in the first claim, they have not mentioned to what extent the feature design affects the performance of NIDS. or even mention which features they were meant for. in the second claim, they have not mentioned the amount of time that feature design takes.

They used an ISCX2012 dataset to perform some experiments to evaluate the model. They compared their method with other methods. They described very well how they set up the environment and mentioned the framework used and the main parameters they used. They mention the evaluation metrics they have used to perform a model evaluation based on it. which are: accuracy, detection rate, false alarm rate, and F1-Score.

They mentioned the reason why they used the ISCX2012 dataset. They used the data of 7 days in which 4 types of attacks and the data from one day to define the normal packets. They mentioned that they used the train test split method to create their train and test datasets.

They had initially supposed that the performance of classification algorithms would vary when the amount of data was small and near each other when the amount of the data was large. They confirmed their assumption through tables where all three classification algorithms gained a fairly steady1-score in the range between 0.90 and 0.97 when the training data size was over 3,200 (1% or more is used as training data). The F1-score fluctuates from 0.72 to 0.94 when the training data is smaller.

As a result of the experiment in [4]

SVM: It's becoming more popular because of its promising performance in prediction and classification. The results show a strong performance: "**UNSW-NB15"** dataset accuracy:**93.75%,** "**CICIDS2017"** dataset accuracy: **98.92%,** "**NSL-KDD"** dataset accuracy**:99.35%,** and "**Kyoto 2006+"** dataset**: 98.58%** Although the training time is high, accuracy has reached more than 98% at least in all four datasets, which is promising. Also, it's more useful when dealing with nonlinear data. On the other hand, it was noticed that Single SVM has a higher False Acceptance Rate

Naïve Bayes accuracy has achieved **89.24%** on the "**NSL-KDD"** dataset**.** The results also show the following: Naïve Bayes is the best on real-time datasets and surprisingly accurate on small data sets rather than big data sets while there is no scaling up of precision, especially in decision trees. The method that has used a hybrid layer with Normalization and transformation operations on the same data set has achieved high accuracy and a low false positive rate.

The result of the **KNN** method showed an effective detection of anomalies with a low false positive rate in the "**Kyoto University Benchmark"** dataset and the "**KDD'99"** dataset.

Logistic Regression performs high accuracy and good interpretability with low computational power. in addition, well-calibrated prediction probabilities are produced without requiring scaling up or tuning input features. Also, Logistic regression outperforms other classifiers because of being more tolerant of feature correlation which makes better predictions.

Random Forest achieved a low false acceptance rate and high DR and accuracy for the "**NSL-KDD"** dataset was **99%** while for the "**KDD cup"** dataset **99.65%** outperformed **LDA** and **CART** with **98.1%** and **98%** in order. Results have shown that more randomness is added to the model by increasing the number of trees. As a result, a lot of variation happens which makes a better model**.** In addition, Random Forest has very good versatility which is appealing in most of its features. One of the most common problems in this model is overfitting, but it would not have occurred if there were enough trees in the forest.

6. Strengths and weaknesses in Available Research

In [1] and [4], it is shown that the strength of these solutions that rely on machine learning is that they offer competitive performance when the training data is relatively small, while the challenge is that a variety of feature engineering and feature augmentation techniques should be applied to the features for ML classifiers to give the desired performance.

In [2] and [3] it is shown that the strength of these solutions that rely on deep learning is that they can learn better feature representation using deep learning algorithms such as CNN, the features learned by deep learning provide better

performance than the manually engineered features in the above solutions while the challenge is that deep learning requires a lot of training data and training time to learn an accurate representation of the features.

7. Gaps in Available Research

In [2], [3] the authors have mentioned the approach they took to handle the problem of unequal numbers of words in each row in the dataset. **But** they have not mentioned the distribution of the number of words in each row.

8. Contrasting Opinions and Missing Claims in Available Research

In [2] Several experiments were performed using different algorithms but with changed feature representations, it would have also been beneficial to see the performance of those algorithms of the chosen CNN feature representation, evaluation metrics relied mostly on the accuracy, and false acceptance rate, but missed some like F1 score as the system could be vulnerable to not detecting a certain type of attack more than the other.

9. Conclusions and proposed solutions

In terms of the training features, the experiments that have been done in [1], [2], [3], and [4] proved that it is better to use deep learning techniques to learn features from data than to prepare it manually.
So, as a proposed solution we will try to improve the network intrusion detection system by using the "KISTI+IDS2021-CDMC " dataset which consists of 4 columns, id, category, word vector of payload embedding, and label either normal or malicious, it contains 123040 rows, and the data is balanced between normal and malicious traffic. **The first step** would be to split the data using stratified K-Fold such that the split data will be balanced, **then** we will **normalize** the word vector length as not all the packets have the same number of words, and we will use the same approach as [2] and [3] to handle the problem of the unequal length of words in each row in the dataset. **Second**, we will experiment with dimensionality reduction techniques such as PCA on the normalized word vector to reduce the number of features while still retaining all the information. **Third**, we will pass the features to a CNN model for it to learn the feature representation needed to separate the classes, we will **then** experiment with feeding this feature representation to many machine and deep learning models to compare their performance.
As data is balanced, both the accuracy and F1 score will be used as evaluation metrics to see if the models will be biased towards the detection of one class better than the other one.

References

[1] X. Zhuo, J. Zhang and S. W. Son, "Network intrusion detection using word embeddings," *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 4686-4695, doi: 10.1109/BigData.2017.8258516.

[2] Erxue Min, Jun Long, Qiang Liu, Jianjing Cui, Wei Chen, "TR-IDS: Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest", *Security and Communication Networks*, vol. 2018, Article ID 4943509, 9 pages, 2018. https://doi.org/10.1155/2018/4943509

[3] Cui, Jianjing, et al. “WEDL-NIDS: Improving Network Intrusion Detection Using Word Embedding-Based Deep Learning Method.” Modeling Decisions for Artificial Intelligence, Springer International Publishing, 2018, pp. 283–95, https://doi.org/10.1007/978-3-030-00202-2_23.

[4] Kocher, G., Kumar, G. Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Comput* **25**, 9731–9763 (2021). https://doi.org/10.1007/s00500-021-05893-0