Moustakidis and Karlsson *Cybersecurity* (2020) 3:16
https://doi.org/10.1186/s42400-020-00056-4

Cybersecurity

RESEARCH Open Access

A novel feature extraction methodology using Siamese convolutional neural networks for intrusion detection

Serafeim Moustakidis* and Patrik Karlsson

Abstract

Intrusion detection systems (IDS) can play a significant role in detecting security threats or malicious attacks that aim to steal information and/or corrupt network protocols. To deal with the dynamic and complex nature of cyber-attacks, advanced intelligent tools have been applied resulting into powerful and automated IDS that rely on the latest advances of machine learning (ML) and deep learning (DL). Most of the reported effort has been devoted on building complex ML/DL architectures adopting a brute force approach towards the maximization of their detection capacity. However, just a limited number of studies have focused on the identification or extraction of user-friendly risk indicators that could be easily used by security experts. Many papers have explored various dimensionality reduction algorithms, however a large number of selected features is still required to detect the attacks successfully, which humans cannot intuitively or immediately understand. To enhance user's trust and understanding on data without sacrificing on accuracy, this paper contributes to the transformation of the available data collected by IDS into a single actionable and easy-to-understand risk indicator. To achieve this, a novel feature extraction pipeline was implemented consisting of the following components: (i) a fuzzy allocation scheme that transforms raw data to fuzzy class memberships, (ii) a novel modality transformation mechanism for converting feature vectors to images (Vec2im) and (iii) a dimensionality reduction module that makes use of Siamese convolutional neural networks that finally reduces the input data dimensionality into a 1-d feature space. The performance of the proposed methodology was validated with respect to detection accuracy, dimensionality reduction performance and execution time on the NSL-KDD dataset via a thorough comparative analysis that demonstrated its effectiveness (86.64% testing accuracy using only one feature) over a number of well-known feature selection (FS) and extraction techniques. The output of the proposed feature extraction pipeline could be potentially used by security experts as an indicator of malicious activity, whereas the generated images could be further utilized and/or integrated as a visual analytics tool in existing IDS.

Keywords: Feature extraction, Siamese convolutional neural networks, Machine learning, Intrusion detection

Introduction

An IDS is a security tool that collects information from various sources (e.g. routers, computers, network data) aiming at identifying malicious activities and/or users that attempt to either get access to computers, steal protected data or even manipulate and disable information systems (Sharma and Gupta 2015). IDSs can be categorized into three main categories (Bijone 2016). The first category of IDS compares the collected patterns of network traffic with specific and pre-determined signatures (attack patterns). An attack is detected once there is match with an already known pattern, however this kind of IDS is incapable of identifying new (unknown) malicious activities. The second category builds on a set of rules and thresholds (specifications) that have been manually specified by security experts. These specification-based IDSs do not generate false alarms when unusual (but legitimate) program behaviors are encountered but in general the specifications development is a tedious and expensive process while the specified set of rules is often very difficult to

* Correspondence: s.moustakidis@aideas.eu
AIDEAS OÜ, Narva mnt 5, Tallinn, Harju maakond, Estonia

evaluate and verify. Unlike signature and specification IDSs, automated intrusion detection (AID) systems is a new category that employs machine learning, statistical-based or knowledge-based methods to define a normal model of the behavior of a computer system. The effectiveness of AID systems depends a lot on the quantity as well as quality of the network traffic patterns that are used as data instances during their training.

In the last few decades, ML has been used to improve intrusion detection (Sahasrabuddhe et al. 2017). There is a large number of related studies using various synthetic datasets (such as KDD-Cup 1999 (http://kdd.ics.uci.edu/databases/kddcup99/) or DARPA 1999 datasets (https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset)) to develop and validate ML-empowered AID systems. Any significant deviation between the observed 'normal' behavior can be regarded as an anomaly, which can be then interpreted as an intrusion. The main assumption of the aforementioned approaches is that malicious behavior differs from typical user behavior. One simplistic method to decide whether a behavior is normal or abnormal is by comparing it with the standard deviation of the normal user behaviors in the training dataset. Any example exceeding the predetermined threshold (e.g. three times the standard deviation) could be classified in the intrusion category. ML provides a more sophisticated method for decision making overcoming the deficiencies of the heuristic approaches (such the manual selection of the threshold etc.). Development of ML-based AID systems comprises of two phases: the training phase and the testing phase.

a. In the training phase, the normal traffic profile is used to learn a model of normal behavior,
b. In the testing phase, a new data set is used to validate the system's capacity to generalize to previously unseen intrusions.

AIDS can be classified into a number of categories based on the method used for training, for instance, statistical based, knowledge-based and machine learning based (Butun et al. 2014). The main advantages of ML-empowered AID systems are: (i) Their ability to identify zero-day attacks without relying on a signature database (Alazab et al. 2012). A danger signal can be triggered when the examined behavior differs from the usual behavior. (ii) Their capability to discover internal malicious activities. An alarm will be created in cases where an intruder starts making transactions in a compromised account that deviate from the typical user activity. (iii)The normal user behavior is hidden to intruders and thus it becomes more difficult for them to remain undetected. The objective of using machine learning techniques is to create IDS with improved accuracy and less requirement for prior human knowledge. However, one of the main challenges of current AIDS is the high false positive rates because anomalies may just be new normal activities rather than genuine intrusions.

One of the crucial phases in today's ML pipelines is the process of extracting knowledge from large quantities of data. To effective extract knowledge from raw data, ML relies on a set of rules, methods, or complex "transfer functions" that are applied to find interesting data patterns, or to recognize and predict behavior (Dua and Du 2016). Many ML algorithms (such as clustering, neural networks, association rules, decision trees, genetic algorithms, and nearest neighbor methods) have been recently applied in the area of AIDs for discovering knowledge from intrusion datasets (Kshetri and Voas 2017; Xiao et al. 2018). Some prior research in data mining has examined the use of different algorithms to extract meaningful information for intrusion data. Two feature selection algorithms were investigated by Chebrolu et al. 2015 employing Bayesian networks (BN) and Classification Regression Trees. The outputs of the aforementioned algorithms were finally combined to increase accuracy. Bajaj and Arora 2013 proposed a technique for feature selection using a hybrid approach that combines Information Gain and correlation attribute evaluation. To validate the discrimination capacity of the selected features, the authors applied several classification algorithms such as C4.5, naïve Bayes, NB-Tree and Multi-Layer Perceptron (Khraisat et al. 2018). Genetic-fuzzy rule mining has been also explored to evaluate the importance of IDS features by Elhag et al. 2015. Thaseen and Kumar 2013 proposed a Random Tree model to improve the accuracy and reduce the false alarm rate, whereas Subramanian et al. 2012 also studied the performance of decision tree algorithms on the NSL-KDD dataset (https://www.unb.ca/cic/datasets/nsl.html). Dimensionality reduction using Principal Component Analysis (PCA) has been also explored to remove noisy attributes and retain the optimal attribute subset towards the development of more accurate and computationally efficient IDS. Thaseen and Kumar 2014 developed an intrusion detection model using PCA as the dimensionality reduction technique and SVM as the classifier, whereas Kuang et al. 2014 adopted a similar approach using PCA for identifying the primary features of an intrusion detection dataset that were used as inputs in a Genetic Algorithm (GA)-assisted support vector machines classifier. Chi-square-based feature selection (Thaseen and Kumar 2017; Thaseen et al. 2018) was finally adopted to rank the available features based on their statistical significance test and finally select only those features that are dependent on the class label.

Unlike ML approaches that require the extraction of features, Deep learning (DL)-based detection methods learn features automatically in an end-to-end fashion (directly from raw data to decisions). DL is gradually attracting more interest in AID studies. An intrusion detection method based on convolutional neural networks (CNN) was proposed by Zheng 2020 in which a three-layer CNN was trained on the KDD99dataset. A CNN-based AID methodology was also presented by Potluri et al. 2018 conducting experiments on the NSL-KDD and the UNSW-NB datasets. In the pre-processing phase, the features of the datasets were transformed into images of 8*8 pixels. Then, a three-layer CNN was trained to classify the attacks. Pre-trained deep networks (ResNet 50 and GoogLeNet) were also explored as alternative solutions to the task of extracting new informative features. The proposed CNN was the best performing approach, reaching accuracies of 91.14% on the NSL-KDD and 94.9% on the UNSW-NB 15. A sparse autoencoder was also proposed by Zhang et al. 2018 to extract features from the NSL-KDD dataset. The extracted features were supplied to an XGBoost model with the objective to detect attacks. To overcome the observed data imbalance problem, data resampling was employed (using SMOTE). The SMOTE algorithm oversamples the minority classes and divides the majority classes into many subclasses so that every class is balanced. Data augmentation with generative adversarial networks (GANs) has been also explored by Zhang et al. 2019. The GAN model was used to generate data similar to the flow data of KDD99. Adding this generated data to the training set increased the generalization capacity of the detection model that was able to identify not only attacks but attack variants as well. Finally, to model the role of queries for role-based access control of databases, a hybrid method (Bu and Cho 2020) combined conventional learning classifier system with CNNs outperforming conventional ML approaches in insiders' attacks detection.

Most of the reported effort in the literature so far has been devoted on building powerful and complex IDS adopting a brute force approach towards the maximization of their detection capacity via the use of ML and/or DL. Despite the availability of many sophisticated detection techniques, only a few papers focus on: (i) identifying the characteristic features for the normal traffic and the attack traffic and (ii) extracting user-friendly risk indicators that could be easily used by security experts. Data and models transparency has been recognized as a critical issue especial when ML/DL approaches are employed in problem domains like cybersecurity, in which human users might like to understand how a given system made a given decision (Marcus 2018). As reported above, some papers have explored various dimensionality reduction algorithms, however a large number of selected features is still required to detect the attacks successfully. This paper contributes to current AID systems by transforming the available multi-dimensional network traffic data into a single actionable and easy-to-understand indicator for security experts. This has been achieved by designing a novel feature extraction pipeline that builds on the latest advances of data mining and ML/DL. Specifically, a fuzzy allocation scheme is initially applied to transform raw data to fuzzy class memberships. Then a data transformation mechanism converts feature vectors to images (Vec2im) and a dimensionality reduction module makes use of Siamese convolutional neural networks to reduce the input data dimensionality into a 1-d feature space. The performance of the proposed methodology was validated via a thorough comparative analysis that demonstrated its effectiveness over a number of well-known feature selection and extraction techniques.

The rest of this article is structured as follows: The proposed feature extraction pipeline with architectural and implementation details is provided in Methodology section. Experimental results are demonstrated in Results section, with a comparative analysis with existing approaches. Conclusions are drawn in Conclusions section.

Methodology

The proposed methodology in this paper for feature extraction in the domain of intrusion detection includes four processing steps: (i) data pre-processing making use of a fuzzy allocation scheme to convert raw data into fuzzy values, (ii) a modality transformation technique that generates images comprising of fuzzy memberships, (iii) a novel feature extraction algorithm employing Siamese convolutional neural networks and finally (iv) a learning process for training, and evaluation of the results, as illustrated in Fig. 1. The proposed framework implements a modality transformation where non-interpretable feature vectors are being transformed to interpretable images that are further processed by Siamese convolutional neural networks. The proposed fuzzification scheme facilitates the creation of meaningful visual patterns in the generated images. The proposed methodology is thoroughly presented in the following sections.

Dataset description

To validate the performance of the proposed feature extraction methodology, the NSL-KDD dataset was employed. NSL-KDD is actually a variant of the KDD99 dataset that is the most widespread IDS benchmark dataset at present. Overcoming the limitations of KDD99 (severely unbalanced data, many duplicated and redundant records), NSL-KDD represents a more balanced dataset with a moderate number of records that has allowed the application of various IDS in a large number of papers whose results are consistent as well as comparable. For the aforementioned reasons, we used it as a benchmark in our analysis.

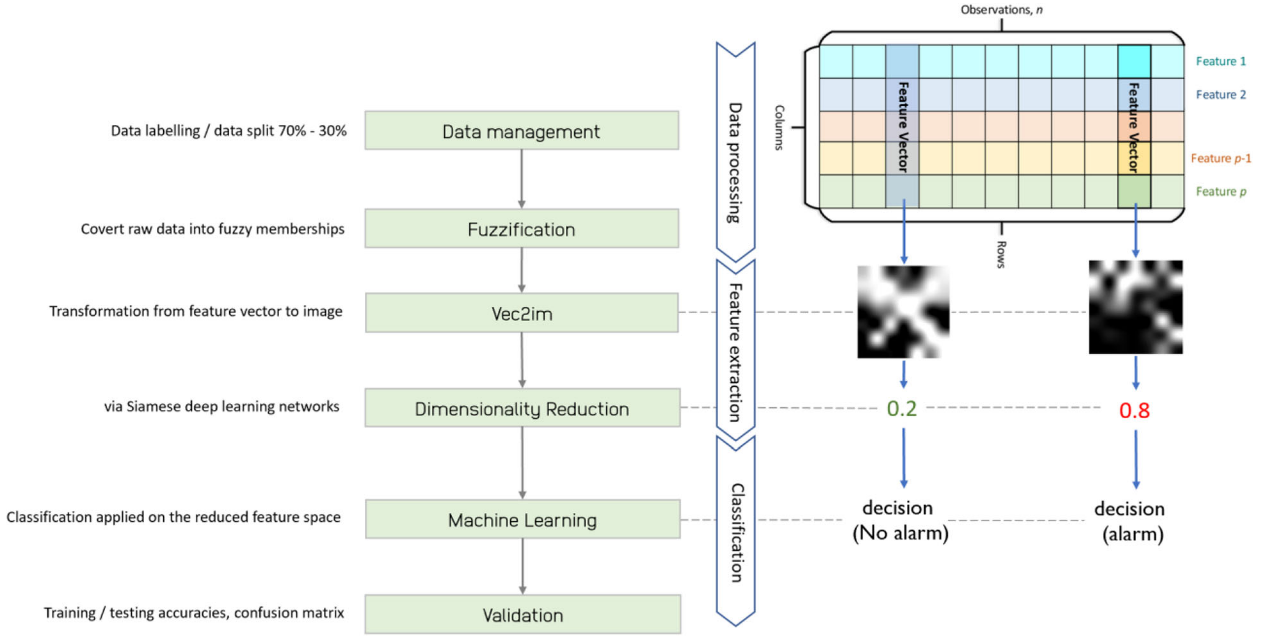

Fig. 1 The proposed Vec2im-Siam methodology

Our dataset consists of 41 features that are categorized in four subsets, i.e., basic features, content features, host-based statistical features, and time-based statistical features. As far as the attack types in the NSL-KDD dataset, they are divided into four categories:

1. **Denial of service** (DoS): exhausting the resources of the attacked object by savage means; thus, making it unable to provide normal services; paralysis. Subcategories: ping of Death, LAND, neptune, backscatter, smurf, teardrop
2. **R2L**: unauthorized access to remote computers. Subcategories: ftp-write, password guessing, imap, multi-hop, phf, spy, warezclient, warezmaster
3. **U2R**: unauthorized access to local superuser privileges. Subcategories: buffer Overflow, loadmodule, perl, rootkit
4. **Probe**: monitoring and other detection behavior. Subcategories: ipsweeping, nmap, portsweeping, satan

Specifically, a 20% subset of the NSL-KDD training data (the NSL-KDD Train 20 variant) was used in our paper comprising of 25,192 data points, where the NSL-KDD Test+ data file (comprising of 22,544 data points) was utilized for testing. Given that the focus of the paper is not on the recognition of the different attacks, the intrusion detection problem was considered as binary by merging all the anomalous records (categories 1–4) into one class.

Fuzzification and image formulation (Vec2im)

First of all, the non-numeric attributes of the dataset were converted into numeric values. For the efficient training of machine learning algorithms, input data is typically transformed by a number of pre-processing routines with data normalization being the gold standard. Different algorithms could be used to normalize the input data (such as min-max normalization or normalization with respect to standard deviation), however in this paper we employed a fuzzy allocation scheme as described below.

Fuzzification

To normalize as well as evaluate the classification capabilities of each feature, we applied a simple fuzzy allocation scheme that assigs varying degrees of patterns to every class. For feature j, the fuzzy membership $u_i(x_{k,j}) \in [0, 1]$ indicating the degree to which $x_{k,\ j}$ belongs to class i is determined by:

$$u_i\left(x_{k,j}\right) = \frac{1}{\sum_{m=1}^{c}\left[\frac{\left(x_{k,j}-u_{i,j}\right)^2}{\left(x_{k,j}-u_{m,j}\right)^2}\right]^{1/(b-1)}} \tag{1}$$

where $u_{i,j} = \sum_{k \in A_i} x_{k,j} / N_i$ is the class i mean along the $x_{k,\ j}$ component, A_i is the set of indexes of the training examples belonging to class i, N_i is the number of class i patterns and b is a fuzzification factor (b = 2 in our experiments). In our paper, every feature component of $x_{k,\ j}$, was converted to $u_1(x_{k,\ j})$ that denotes its fuzzy membership to class 1 (normal / non intrusion class). High values of $u_1(x_{k,\ j})$ close to 1 indicate a strong membership to the non-intrusion class whereas low $u_1(x_{k,\ j})$ values close to 0 are representative of examples belonging to the malicious class.

The motivation for employing the proposed fuzzification mechanism is based on the following remarks: (i) Compared to the typically used normalisation techniques, the proposed fuzzification not only scales features to a common data range but also provides insights with respect to the discrimination capacity of each feature; (ii) Transforming raw into fuzzy values forms a first level of analysis that enables security experts to better understand the feature dynamics (how close the feature value is to the class centres); (iii) The generated fuzzy memberships facilitate the formulation of more interpretable feature images (refer to Vec2im that is described in the following paragraphs) resulting to homogenised visual patterns that could be easily seen/spotted by the security experts.

Vec2im

At the second phase of processing, the generated memberships were re-placed in a matrix format resulting to one grey-scale image per example. Specifically, the 41 features $x_{k,\ j}$, $j = 1, \ldots, 41$ were transformed to 41 fuzzy memberships $u_i(x_{k,\ j})$, $j = 1, \ldots, 41$ and finally a 7×7 image was created per sample by placing the fuzzy memberships in a matrix as presented in Fig. 2. Zero values were also included in the matrix in random cells to fill the eight gaps (given that the dimensionality of the initial feature set was 41 with a total of 49 cells to be filled in the matrix). Fuzzy memberships and zero values were ordered randomly since it was concluded that their order has not any significant impact on the final performance of the proposed methodology.

The rationale behind the proposed Vec2im is threefold: (i) Vec2im implements a modality transformation where non-interpretable feature vectors are being transformed to the image domain. Looking at these visual representations of data, a human can extract relevant information and a sense of the meaning it conveys. (ii) DL has been recently proven to be extremely successful in various domains especially the ones involving images. Transforming the feature modality to the visual domain enables the application of a large number of powerful CNN architectures including pre-trained ones bypassing the requirement of large amounts of labeled data to facilitate the training of these very deep networks. (iii) The combination of the proposed fuzzification and Vec2im mechanisms lead to easy-to-understand by the users images in which white areas (fuzzy memberships close to 1) represent features that strongly belong to the normal traffic class and vice versa. Using the generated fuzzy memberships (instead of raw feature values) contributes to the generation of distinguishable visual patterns that can be captured by the Siamese convolutional neural network that follows.

Dimensionality reduction with Siamese deep learning networks

Deep Siamese Convolutional Neural Networks (SCNN) architecture is a variant of neural networks that was originally designed to solve signature verification problem of image matching (Bromley et al. 1993). It has also been used for one-shot image classification (Koch 2015), face verification where the categories are not known in advance (Chopra et al. 2005) as well as for dimensionality reduction (Hadsell et al. 2006). SCNN consist of two identical symmetric CNN subnetworks that share the same weights. In our experiment, each identical CNN was built using one convolutional layer followed by three fully connected layers. The rectified linear units (ReLU) nonlinearity was applied as the activation function for all layers, and adaptive moment estimation (ADAM) optimizer was utilized to control learning rate (Kingma and Ba 2014). The similarity between images was calculated by Euclidean distance, and the contrastive loss (Chopra et al. 2005) was calculated to define the loss function as follows:

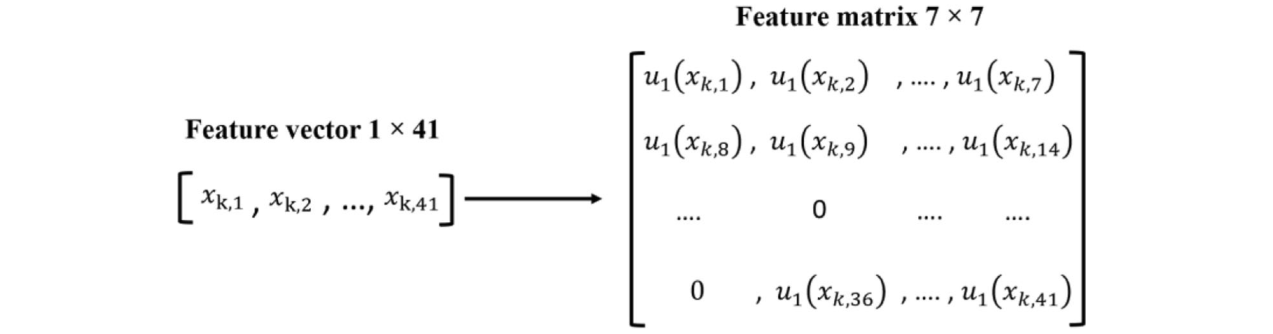
Feature vector 1 × 41
$[x_{k,1}, x_{k,2}, \ldots, x_{k,41}]$
Feature matrix 7 × 7
$u_1(x_{k,1}), u_1(x_{k,2}), \ldots, u_1(x_{k,7})$
$u_1(x_{k,8}), u_1(x_{k,9}), \ldots, u_1(x_{k,14})$
.... 0 ....
0, $u_1(x_{k,36}), \ldots, u_1(x_{k,41})$

Fig. 2 Converting feature vectors to images (Vec2im)

$$\mathscr{L}(W,\ I_1,\ I_2) = \mathbf{1}(L=0)\frac{1}{2}D^2 + \mathbf{1}(L=1)\frac{1}{2}[\max(0,\ margin\text{-}D)]^2 \tag{2}$$

$$\text{where } D = \|f(I_1) - f(I_2)\|^2, \tag{3}$$

$I_1$and I_2 are a pair of the generated images fed into each of two identical CNNs. 1(·) is an indicator function to show that whether two images have the same label, where L = 0 represents the images have the same label and L = 1 represents the opposite. W is the shared parameter vector comprising of the weights that both neural networks share each other. $f(I_1)$ and $f(I_2)$ are the latent representation vectors of input I_1 and I_2, respectively and D is the Euclidean distance between them. The selected SCNN architecture (as depicted in Fig. 3) reduces the dimensionality of the 41-dimensional feature space to a single 1-d space.

Decision making on the reduced feature space

To evaluate the discrimination capacity of the extracted features, we employed various machine learning models trained to implement the binary classification task on the resulted 1-d space. We tested linear discriminant analysis (LDA) and Naïve Bayes (Duda et al. 2000) to provide a baseline for comparisons with more advanced models. We also evaluated decision trees (Belson 1959; Witten et al. 2011), driven by Gini's diversity index, KNN (Atkeson et al. 1997), as well as non-linear support vector machines (SVM) algorithms (Cortes and Vapnik 1995; Scholkopf 1997) with Gaussian kernel, which can deal with the overfitting problems that appear in high-dimensional spaces. The ensemble techniques AdaBoost (AB) (Freund and Schapire 1997) and Random Forest (RF) (Breiman 2001) were also evaluated using decision trees (DT) models as weak learners.

To achieve a fair comparison between the different approaches, hyperparameter selection was performed for each one of the investigated machine algorithms. A validation subset was held out from the training set (a randomly selected 10%) as a criterion for: (i) selecting the optimum hyperparameters by means of a grid search process as well as (ii) deciding the termination of the SCNN learning.

Results

Data visualization

Figure 4 depicts indicative images generated by the proposed Vec2im for both intrusion and non-intrusion (normal) classes. White areas correspond to features in which a strong membership to the normal class is observed and vice versa for the black areas. Observing the generated images, there is a clear visual distinction between the two classes, with the normal class (Fig. 3a) being represented by whiter images. Some of the pixels receive high values (close to 1) in images from both classes, however there are some areas on the images that are solely activated in one of the two classes creating distinct color patterns. The objective of the Siamese neural network that follows is to capture these patterns via their convolutional layer and further convert this information into a more compressed representation (reduced feature space).

Siamese network learning

Figure 5a shows the progression of contrastive loss of the SCNN with respect to the number of iterations. One critical aspect of SCNN training is the termination of the learning process. A validation subset was held out from the training set (a randomly selected 10%) and a linear classifier (LDA in our paper), trained on the 90% of the training set, was utilized to act as a termination

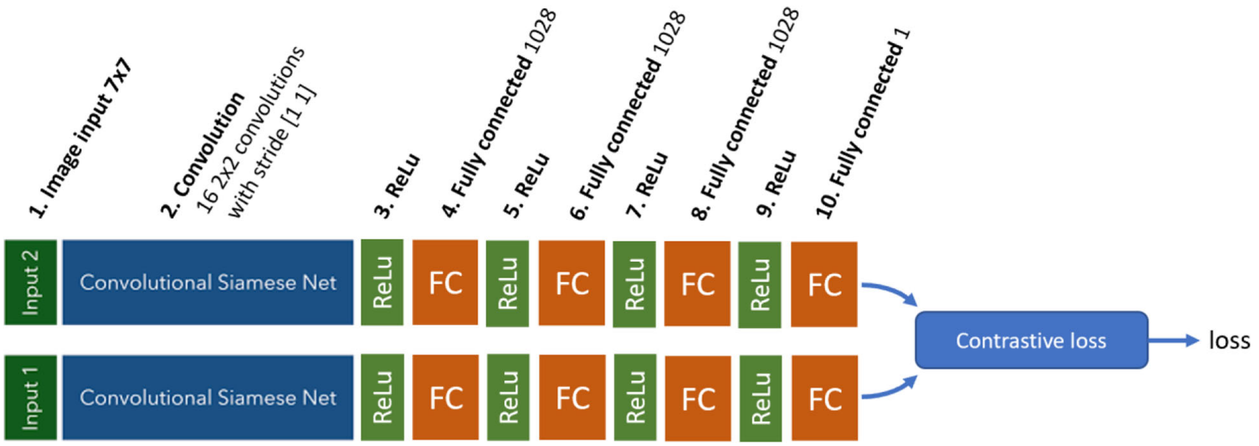

Fig. 3 Selected Siamese network architecture for dimensionality reduction

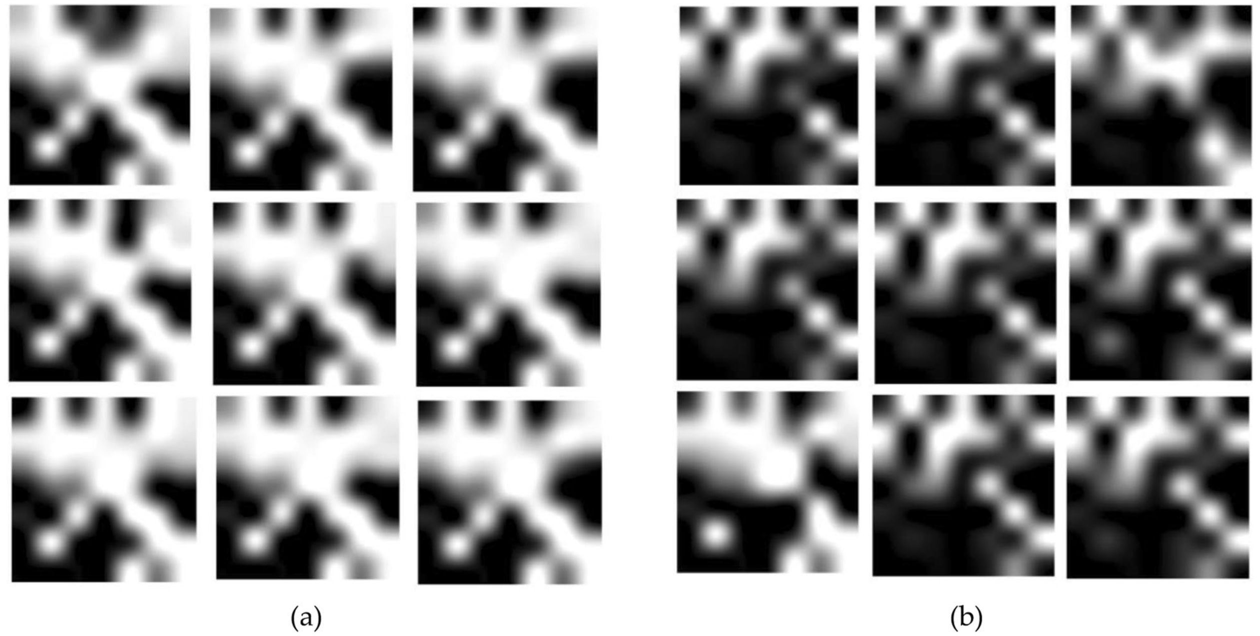

Fig. 4 Indicative images generated by Vec2im for the normal (**a**) and intrusion class (**b**)

criterion. Specifically, the learning process terminates on the iteration where the validation performance of the trained LDA models reaches its maximum value (at iteration 441 in our example). Figure 5b and c depict the histogram of the extracted feature values on the reduced 1-d space (that is actually the SCNN output) at iterations 1 and 441 iterations, respectively. The histogram at the first iteration shows that there is a significant overlap between the data distribution of the two classes. On the contrary, the resulted space at iteration 441 is more informative with a small overlap between the per-class distributions (Fig. 5c) and with most of data points concentrated at the distribution edges.

Comparative analysis

Identifying the optimum ML performer on the reduced feature space

Seven ML models were investigated for their suitability on discriminating intrusion from non-intrusion data. Specifically, we tested: (i) Naïve Bayes, (ii) AdaBoost, (iii) Random Forest, (iv) Support Vector Machines, (v) nearest neighbor classifier (kNN), (vi) decision trees and (vii) discriminant analysis trained on the reduced 1-d feature space as have been generated by Vec2im-Siam.

Figure 6 shows the progression of the testing accuracy in relation to the number of iterations for the aforementioned ML models. To implement this, we stored the extracted

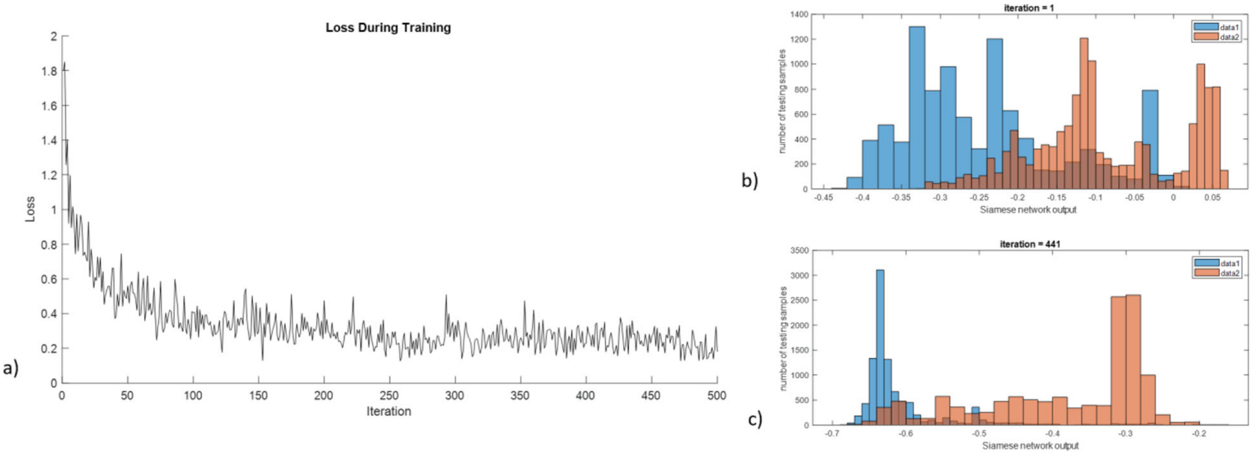

Fig. 5 **a**) Training contrastive loss with respect to number of iterations, **b**) histogram of the reduced feature space (SCNN output) at iteration 1 and **c**) histogram of the reduced feature space (SCNN output) at iteration 441

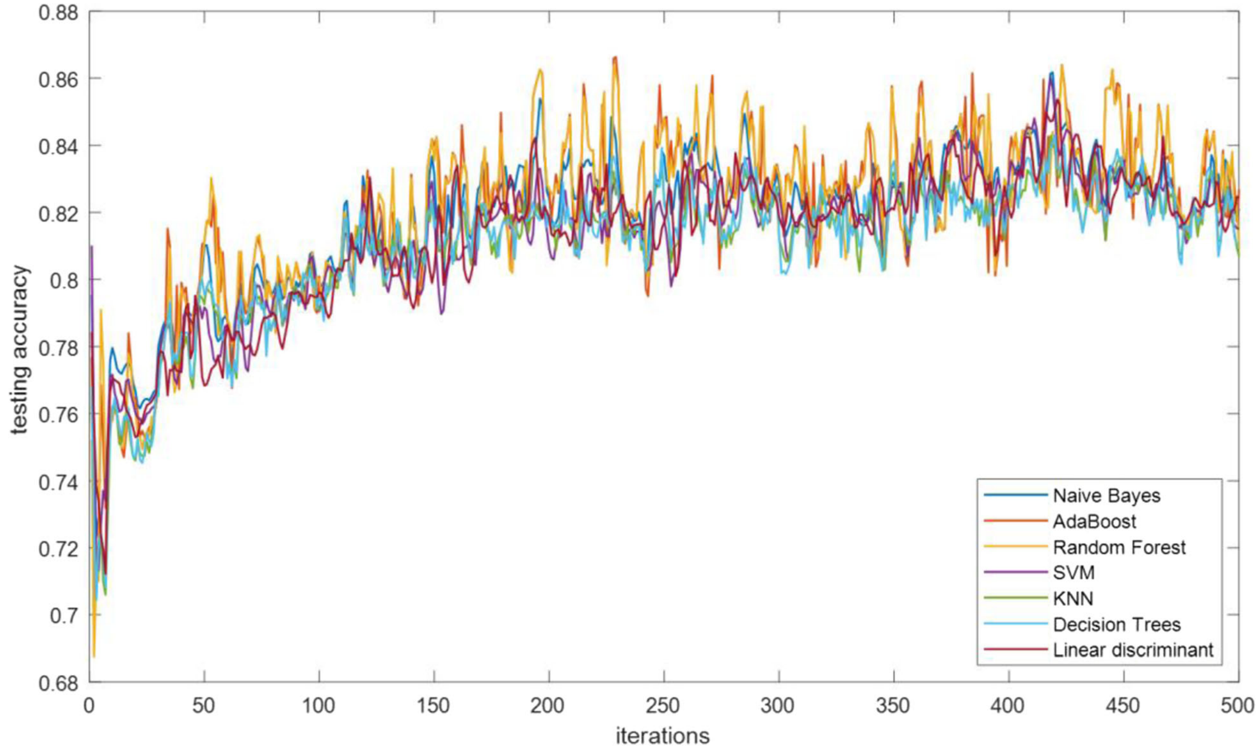

Fig. 6 Testing accuracy with respect to number of iteration of different ML models trained on the resulted 1-d feature space

feature values for both training and testing after the end of each iteration of the SCNN learning process. This led to the creation of 500 1-d training and testing sets in which the seven ML were trained and validated, respectively.

Table 1 cites the best performances (training and testing) as accomplished by the seven competing ML models. AdaBoost achieved the overall best testing accuracy (86.64%) whereas slightly reduced testing accuracies (more than 86%) were received by RF, Naïve Bayes and SVM. LDA, DT and kNN were 1% - 2% less effective in our data classification task. Overall, all competing models had similar learning curves (as shown in Fig. 6) and they achieved similar testing accuracies within a range of approximately 2%. This finding verifies the effectiveness of the proposed feature extraction methodology that leads to a very informative 1-d feature space in which all ML models (either linear or non-linear) perform well. The confusion matrixes of the four best performing ML models (as shown in Table 2) demonstrate that all four perform similarly exhibiting non important differences in the achieved class accuracies.

Comparison with other feature selection / feature extraction techniques

A thorough comparative analysis between the proposed methodology and other well-known competing feature extraction / selection techniques is presented below.

- ***Experiments 8–9***: *Evaluating the effect of fuzzification on the proposed feature extraction methodology*

In experiments 8–9, we evaluated the effect of fuzzification on the performance of the proposed methodology. To accomplish this, we replaced the proposed fuzzification technique with a standard data normalization into the range [0, 1]. Both pre-processing techniques scale the data into the same range, however they have different characteristics:

- data normalization applies a linear transformation on the data rescaling all features to the same range, whereas
- the proposed fuzzification technique transforms raw data into class memberships (the generated fuzzy values declare at what extend a sample belongs to class 1).

The application of the fuzzy allocation scheme prior to feature extraction had a positive effect on both training and testing accuracies. Specifically, replacing the proposed fuzzification scheme with a standard data normalization technique lowered the testing accuracy by 6% (refer to experiment 9 in Table 3).

- ***Experiments 10–28***: *Comparing the proposed feature extraction approach with other FS techniques*

Table 1 Comparative analysis with respect to optimal choice of the machine learning model

Exp.	Feature Extraction	Classification model	Training (%)	Testing (%)
1.	Fuzz-Vec2im -Siam	Naïve Bayes	98.52	86.17
2.	Fuzz-Vec2im -Siam	AdaBoost	98.43	**86.64**
3.	Fuzz-Vec2im -Siam	Random Forest	98.48	86.40
4.	Fuzz-Vec2im -Siam	SVM	98.53	86.01
5.	Fuzz-Vec2im -Siam	kNN - 1	**100**	84.29
6.	Fuzz-Vec2im -Siam	Decision Trees	99.5	84.31
7.	Fuzz-Vec2im -Siam	Linear Discriminant	98.69	85.38

For comparison purposes, a wrapper FS technique (Wr-FS) was also employed to reduce the feature dimensionality of the initial 41-d space. This technique employs a search strategy to look through the space of possible feature subsets, evaluating each subset based on the quality of the performance of a given algorithm. A sequential forward selection strategy was implemented that starts with no feature and progressively adds one feature at a time. The same classifier was utilized in both the wrapper FS and the proposed Vec2im-Siam methodology in order to set a fair comparison ground. The wrapper FS technique was implemented four times setting different termination criteria as follows:

- in the experiment 10, the wrapper FS was implemented to identify the most important feature from the entire feature space. The obtained 1-d space was compared with the 1-d space as it has been generated by the proposed in this paper methodology. A much lower testing accuracy was observed in experiment 10 (77.91%) indicating that the selected feature is less informative compared to the extracted feature from Vec2im-Siam.
- In experiments 12, 14 and 16, different termination criteria were set in the Wrapper FS technique leading to feature spaces of higher dimensionality (2-d, 3-d and 10-d, respectively). The discrimination capability of the resulted spaces was also compared with the 1-d space of the proposed methodology using the same classifier. The results verify that the extracted 1-d feature space of Vec2im-Siam is more descriptive than the resulted spaces of the Wrapper FS technique. This is even valid in the case of the 10-d feature space, in which the application of the same classifier led to an testing accuracy of 84.06 (exp.16) that is 2.6% lower than the testing performance of the proposed methodology implemented on a 1-d space.

Table 2 Confusion matrixes of the best four performing ML models trained on the reduced feature space

AB		Class 1	Class 2	Per class accuracy
	Class 1	9323	388	96.00%
	Class 2	2624	10,209	79.55%
		Overall accuracy		**86.64%**
RF		Class 1	Class 2	Per class accuracy
	Class 1	9336	375	96.14%
	Class 2	2692	10,141	79.02
		Overall accuracy		**86.40%**
NB		Class 1	Class 2	Per class accuracy
	Class 1	9393	318	96.73%
	Class 2	2800	10,033	78.18%
		Overall accuracy		**86.17%**
SVM		Class 1	Class 2	Per class accuracy
	Class 1	9400	311	96.80%
	Class 2	2842	9991	77.85%
		Overall accuracy		**86.01%**

To further evaluate the usefulness of the proposed fuzzification routine, we applied it as a pre-processing tool prior to the application of the Wrapper FS technique (refer to experiments 11, 13, 15 and 17) and finally compared the classification accuracy obtained with the one obtained in experiments 10, 12, 14 and 16 (where a standard data normalization is employed). Equal accuracies were obtained in experiments 10 and 11 (77.91% in testing for both) and in experiments 12 and 13 (82.04% in testing for both). A slight increase was observed in the testing accuracy of experiments 15 and 17 (85.81% and 85.93%, respectively) compared to the accuracies achieved in experiments 14 and 16 (82.98% and 84.06%, respectively) indicating that the proposed fuzzification algorithm might also have a positive effect on a variety of other machine learning pipelines.

In experiments 18–28, the performance of the proposed methodology was compared with eleven (11) well known DR techniques of the recent literature. Table 3 cites the selected techniques along with their main characteristics, whereas Table 4 presents their training and testing accuracies on the NSL-KDD dataset employed in

Table 3 Well known DR methods from the literature that were compared with the proposed Vec2im-Siam

DR approach	Acronym	Description
Infinite Latent Feature Selection (Roffo et al. 2017)	ILFS	A probabilistic latent feature selection approach that performs the ranking step by considering all the possible subsets of features bypassing the combinatorial problem
Unsupervised graph-based filter (Roffo et al. 2015)	Inf-FS	In Inf-FS, each feature is a node in the graph, a path is a selection of features, and the higher the centrality score, the most important the feature. It assigns a score of importance to each feature by taking into account all the possible feature subsets as paths on a graph.
Relief-F (Liu and Motoda 2007)	Relief-F	An iterative, randomized, and supervised approach that estimates the quality of features according to how well their values differentiate data samples that are near to each other; it does not discriminate among redundant features, and performance decreases with few data.
Laplacian Score (He et al. 2005)	LS	The importance of a feature is evaluated by its power of locality preserving. In order to model the local geometric structure, this method constructs a nearest neighbor graph. LS algorithm seeks those features that respect this graph structure.
Fisher filter feature selection (Gu et al. 2011, Xue-qin et al. 2006)	Fisher	It computes a score for a feature as the ratio of interclass separation and intraclass variance, where features are evaluated independently, and the final feature selection occurs by aggregating the m top ranked ones.
Correlation-based Feature Selection (Shahbaz et al. 2016)	CFS	CFS sorts features according to pairwise correlations
Unsupervised Feature Selection with Ordinal Locality (Guo et al. 2017)	UFSOL	A clustering-based approach that preserves the relative neighborhood proximities and contributes to distance-based clustering
Least Absolute Shrinkage and Selection Operator (Hagos et al. 2017)	Lasso	This method applies a regularization process that penalizes the coefficients of the regression variables while setting the less relevant to zero to respect the constraint on the sum. FS is a consequence of this process when all the variables that still have non-zero coefficients are selected to be part of the model
Chi-square feature selection (Thaseen and Kumar 2017; Thaseen et al. 2018)	Chi2	It ranks features based on the statistical significance test and consider only those features that are dependent on the class label
Minimum redundancy maximum relevance (Nguyen et al. 2010)	mRMR	A FS algorithm that systematically performs variable selection, achieving a reasonable trade-off between relevance and redundancy.
Fuzzy Complementarity Criterion (Moustakidis et al. 2012, Moustakidis and Theocharis 2010)	FuzCoC	FS is driven by a fuzzy complementary criterion which assures that features are iteratively introduced, providing the maximum additional contribution with regard to the information content given by the previously selected features

this paper. Specifically, each one of the 11 aforementioned approaches returned a feature ranking. LDA was progressively trained on feature subsets of increasing dimensionality starting with the most important feature and adding sequentially one feature at each step (with the feature order determined by each of the 11 ranking techniques). The learning process was performed using the training dataset and the optimal feature subset was the one that maximizes the performance on the validation set. The training and testing accuracies of the optimal feature subset are given in Table 4. Compared to Vec2im-Siam, considerably lower testing accuracies were received by all the competing DR techniques with Lasso (exp 25) achieving the highest performance among the 11 techniques (82.04%), however at a much higher dimensionality (22 features). Overall, the proposed methodology outperformed all the competing techniques both in testing accuracy (more than 4% more accurate) and in the reduction of the initial space dimensionality (only one feature is finally extracted by Vec2im-Siam, whereas more features are required by the rest of the techniques).

- ***Experiments 29–31****: Comparing the proposed feature extraction approach with PCA*

The proposed Vec2im-Siam feature extraction methodology was finally compared with Principal Component Analysis (PCA) that is a well common feature dimensionality reduction approach. Three different feature spaces of varying dimensionality were generated via PCA as follows:

- – A 1-d feature space using only the first extracted principal component (experiment 29)
- – A 2-d feature space using the first two extracted principal components (experiment 30)
- – A 3-d feature space using the first three extracted principal components (experiment 31).

The results highlighted the superiority of the Vec2im-Siam given that it outperformed PCA by more than 10% in testing even in the case in which higher dimensional spaces were used (e.g. in experiments 30 and 31).

Table 4 Comparative analysis with respect to competing feature selection / extraction techniques using the same classifier (LDA)

Exp.	Preprocessing	Feature Selection / Extraction	Dimensionality of the resulted feature space	Training (%)	Testing (%)	Execution time (s)
8.	Fuzzification	Vec2im – Siam	1	**98.69**	**86.64**	113.736
9.	Normalization	Vec2im – Siam	1	90.20	80.64	113.245
10.	Normalization	Wr-FS (best feature)	1	82.84	77.91	1.991
11.	Fuzzification	Wr-FS (best feature)	1	82.84	77.91	1.953
12.	Normalization	Wr-FS (2 first features)	2	90.79	82.04	3.048
13.	Fuzzification	Wr-FS (2 first features)	2	90.79	82.04	3.161
14.	Normalization	Wr-FS (3 first features)	3	92.20	82.98	4.370
15	Fuzzification	Wr-FS (3 first features)	3	91.24	85.81	4.788
16.	Normalization	Wr-FS (10 first features)	10	92.33	84.06	17.091
17.	Fuzzification	Wr-FS (10 first features)	10	92.03	85.93	17.214
18.	Normalization	ILFS	39	95.44	78.44	0.307
19.	Normalization	InfFS	13	93.91	81.94	**0.063**
20.	Normalization	Relief-F	13	92.96	81.75	122.105
21.	Normalization	LS	33	95.42	78.50	158.381
22.	Normalization	Fisher	13	93.91	81.94	0.415
23.	Normalization	CFS	24	93.14	81.51	0.071
24.	Normalization	UFSOL	39	95.44	78.44	312.436
25.	Normalization	Lasso	22	93.61	82.04	14.505
26.	Normalization	Chi2	33	95.44	78.48	1.113
27.	Normalization	mRMR	26	94.70	81.24	1.060
28.	Normalization	FuzCoC	25	94.61	81.94	1.160
29.	Normalization	PCA (1 pr. component)	1	89.69	75.64	0.144
30.	Normalization	PCA (2 pr. components)	2	89.47	76.30	
31.	Normalization	PCA (3 pr. components)	3	89.48	76.32	

Computational analysis The computational efficiency of the proposed and all the competing algorithms was also investigated with respect to training execution time (time in seconds required by each algorithm during the training phase to produce the final feature ranking or extract the reduced feature subset). All algorithms were implemented on the same machine (Intel Core TM – i7–7500 CPU at 2.70GHz) using MATLAB® 2020a. The competing FS algorithms were developed using the FSLib 2018 library (Giorgio 2020). Moderate computational performance (~ 113 s) was achieved by the proposed methodology, whereas InfFS was the fastest algorithm (feature ranking was generated within 0.063 s). UFSOL was proved to be slowest algorithm (implemented in 312.436 s). Overall, the following remarks could be extracted from the comparative analysis of Table 4: (i) Vec2im-Siam was the most accurate in both training and testing (98.69% and 86.64%, respectively); (ii) It achieved the maximum dimensionality reduction performance where only one feature is finally extracted from the initial feature space; (iii) It can be executed in approximately 113 s that is not prohibitive for offline use. Training of such an AID system could be performed periodically at predetermined time periods to potentially include new identified attach patterns. Once constructed, the trained Vec2im-Siam approach could be either used in (almost) real time applications since its actual response time per input has been calculated to be 0.066 s enabling its utilization in real world scenarios (in integrated IDS where immediate responses to identified attacks are required).

Conclusions

A novel feature extraction pipeline is proposed in this paper that consists of the following components: (i) a fuzzy allocation scheme that transforms raw data to fuzzy class memberships, (ii) a mechanism for converting feature vectors to image (Vec2im) and (iii) a dimensionality reduction module that makes use of Siamese convolutional neural networks that finally reduces the input data dimensionality into a 1-d feature space. The proposed methodology was successfully applied on the NSL-KDD intrusion detection dataset. A thorough comparative analysis was performed that demonstrated the effectiveness of the different components of the methodology (e.g. the fuzzification, Vec2im including also visual interpretation) and also compared its performance with other well-known feature selection and extraction techniques. The proposed feature extraction methodology was assessed by applying a variety of ML models on the resulted 1-d feature space and evaluating their performances on the testing dataset. The increased discrimination capability of the resulted reduced feature space was verified by the fact that all competing models that were trained on it (either linear or not) had similar learning curves achieving similar testing accuracies within a small range of approximately 2%. The outcome of the proposed pipeline (Vec2im-Siam) could be potentially used as a risk indicator for identifying cyber attacks, whereas the generated Vec2im images could be further utilized and/or integrated as a visual analytics tool in IDS. Future plans include the application of the proposed methodology in other sectors (such as in healthcare) where decisions should be taken based on complex and heterogenous data.

Abbreviations

IDS: Intrusion detection system; ML: Machine learning; DL: Deep learning; FS: Feature selection; AID: Automated intrusion detection; BN: Bayesian networks; CNN : Convolutional neural network; GAN: Generative adversarial network; Vec2im: Vector to Image; SCNN: Siamese convolutional neural network; ReLU: Rectified linear units; ADAM: Adaptive moment estimation; LDA: Linear discriminant analysis; SVM: Support vector machine; AB: AdaBoost; RF: Random Forest; DT: Decision tree; kNN: Nearest neighbor classifier

Authors' contributions

Serafeim Moustakidis designed the feature extraction pipeline, performed the experiments and drafted the manuscript. Patrik Karlsson participated in problem discussions and improvements of the manuscript. The author(s) read and approved the final manuscript.

Funding

This work has received funding from the European Community's H2020 Programme, under grant agreement Nr. 826183 (SPHINX).

Availability of data and materials

NSL-KDD data can be found at: https://www.unb.ca/cic/datasets/nsl.html.

Competing interests

The authors declare that they have no competing interests.

Received: 20 May 2020 Accepted: 21 July 2020
Published online: 14 August 2020

References

Alazab A, Hobbs M, Abawajy J, Alazab M (2012) Using feature selection for intrusion detection system. In: Communications and information technologies (ISCIT), 2012 international symposium on. IEEE, pp 296–301. http://dro.deakin.edu.au/view/DU:30048268

Atkeson C, Moore A, Schaal S (1997) Locally weighted learning. Artif Intell Rev 11: 11–73

Bajaj K, Arora A (2013) Dimension reduction in intrusion detection features using discriminative machine learning approach. Int J Comput Sci Issues 10(4):324–328

Belson W (1959) Matching and prediction on the principle of biological classification. Appl Stat 8:65

Bijone M (2016) A survey on secure network: intrusion detection prevention approaches. Am J Inf Syst 20(4):69–88

Breiman L (2001) Random forests. Mach Learn 45:5–32

Bromley J, Guyon I, Lecun Y, Sackinger E, Shah R (1993) Signature verification using a Siamese time delay neural network. In: Cowan J, Tesauro G (eds) Advances in neural information processing systems (NIPS 1993), vol 6. Morgan Kaufmann. https://nyuscholars.nyu.edu/en/publications/signature-verification-using-a-siamese-time-delay-neural-network

Bu S, Cho S (2020) A convolutional neural-based learning classifier system for detecting database intrusion via insider attack. Inf Sci 512:123–136. https://doi.org/10.1016/j.ins.2019.09.055

Butun I, Morgera SD, Sankar R (2014) A survey of intrusion detection systems in wireless sensor networks. IEEE Commun Surv Tutor 16(1):266–282

Chebrolu S, Abraham A, Thomas JP (2015) Feature deduction and ensemble design of intrusion detection systems. Comput Secur 24(4):295–307 6

Chopra S, Hadsell R, LeCun Y (2005) Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) 2005, vol 1. IEEE, pp 539–546. https://ieeexplore.ieee.org/abstract/document/1467314

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297

DARPA dataset. https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset. Accessed 15 May 2020

Dua S, Du X (2016) Data mining and machine learning in cybersecurity. CRC press. https://dl.acm.org/doi/book/10.5555/2018783

Duda R, Hart P, Stork D (2000) Pattern classification, 2nd edn. Wiley Inter Science, Hoboken

Elhag S, Fernández A, Bawakid A, Alshomrani S, Herrera F (2015) On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems. Expert Syst Appl 42(1):193–202

Freund Y, Schapire R (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55:119–139

Giorgio (2020) Feature selection library. MATLAB Central File Exchange. https://www.mathworks.com/matlabcentral/fileexchange/56937-feature-selection-library. Retrieved July 3, 2020

Gu Q, Li Z, Han J (2011) Generalized fisher score for feature selection. In: Proceedings of the 27th conference on uncertainty in artificial intelligence, UAI, pp 266–273

Guo J, Quo Y, Kong X, He R (2017) Unsupervised feature selection with ordinal locality. In: 2017 IEEE international conference on multimedia and expo (ICME). IEEE, pp 1213–1218. https://ieeexplore.ieee.org/document/8019357

Hadsell R, Chopra S, LeCun Y (2006) Dimensionality reduction by learning an invariant mapping. CVPR (2). IEEE Computer Society, pp 1735–1742 ISBN: 0-7695-2597-0. https://ieeexplore.ieee.org/document/1640964?section=abstract

Hagos D, Yazidi A, Kure Ø, Engelstad PE (2017) Enhancing security attacks analysis using regularized machine learning techniques. In: 2017 IEEE 31st international conference on advanced information networking and applications (AINA), Taipei, pp 909–918. https://doi.org/10.1109/AINA.2017.19

He X, Cai D, Niyogi P (2005) Laplacian score for feature selection. Adv Neural Inf Proces Syst 18:507–514

KDD Cup 1999. http://kdd.ics.uci.edu/databases/kddcup99/. Accessed 15 May 2020

Khraisat A, Gondal I, Vamplew P (2018) An anomaly intrusion detection system using C5 decision tree classifier. In: Trends and applications in knowledge discovery and data mining. Springer International Publishing, Cham, pp 149–155

Kingma D P, Ba J (2014) Adam: a method for stochastic optimization. CoRR, abs/1412.6980

Koch GR (2015) Siamese neural networks for one-shot image recognition. Thesis

Kshetri N, Voas J (2017) Hacking power grids: a current problem. Computer 50(12):91–95

Kuang F, Xu W, Zhang S (2014) A novel hybrid KPCA and SVM with GA model for intrusion detection. Appl Soft Comput 18:178–184. https://doi.org/10.1016/j.asoc.2014.01.028

Liu H, Motoda H (2007) Computational methods of feature selection. Data mining and knowledge discovery series. Chapman & Hall/CRC, Boca Raton

Marcus G (2018) Deep Learning: A Critical Appraisal, Preprint at http://arXiv:1801.00631v1.

Moustakidis S, Theocharis J (2010) SVM-FuzCoC: a novel SVM-based feature selection method using a fuzzy complementary criterion. Pattern Recogn 43(11):3712–3729. https://doi.org/10.1016/j.patcog.2010.05.007

Moustakidis S, Theocharis J, Giakas G (2012) Feature selection based on a fuzzy complementary criterion: application to gait recognition using ground reaction forces. Comput Methods Biomech Biomed Eng 15(6):627–644. https://doi.org/10.1080/10255842.2011.554408

Nguyen HT, Petrović S, Franke K (2010) A comparison of feature-selection methods for intrusion detection. In: Kotenko I, Skormin V (eds) Computer network security. MMM-ACNS 2010. Lecture notes in computer science, vol 6258. Springer, Berlin, Heidelberg

Potluri S, Ahmed S, Diedrich C (2018) Convolutional neural networks for multi-class intrusion detection system. In: Mining intelligence and knowledge exploration. Springer, Cham, pp 225–238

Roffo G, Melzi S, Castellani U, Vinciarelli A (2017) Infinite latent feature selection: a probabilistic latent graph-based ranking approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1398–1406

Roffo G, Melzi S, Cristani M (2015) Infinite feature selection. In: 2015 IEEE international conference on computer vision (ICCV), Santiago, Chile, 07-13 Dec 2015, pp 4202–4210. ISBN 9781467383912. https://doi.org/10.1109/ICCV.2015.478

Sahasrabuddhe A, Naikade S, Ramaswamy A, Sadliwala B, Futane P (2017) Survey on intrusion detection system using data mining techniques. Int Res J Eng Technol 4(5):1780–1784

Scholkopf B (1997) Support vector learning. R. Oldenbourg Verlag, Munich

Shahbaz M, Xianbin W, Behnad A, Samarabandu J (2016) On efficiency enhancement of the correlation-based feature selection for intrusion detection systems. In: 2016 IEEE 7th annual information technology, electronics and mobile communication conference (IEMCON), Vancouver, BC, pp 1–7. https://doi.org/10.1109/IEMCON.2016.7746286

Sharma S, Gupta R (2015) Intrusion detection system: a review. Int J Secur Its Appl 9:69–76

Subramanian S, Srinivasan VB, Ramasa C (2012) Study on classification algorithms for network intrusion systems. J Commun Comput 9(11):1242–1246

Thaseen I, Kumar C (2013) An analysis of supervised tree based classifiers for intrusion detection system. In: 2013 international conference on pattern recognition, informatics and mobile engineering, pp 294–299

Thaseen I, Kumar C (2014) Intrusion detection model using fusion of PCA and optimized SVM. In: International conference on contemporary computing and informatics (IC3I), Mysore, pp 879–884. https://doi.org/10.1109/IC3I.2014.7019692

Thaseen I, Kumar C (2017) Intrusion detection model using fusion of chi-square feature selection and multi class SVM. J King Saud Univ Comp Inf Sci 29(4): 462–472. https://doi.org/10.1016/j.jksuci.2015.12.004

Thaseen I, Kumar C, Ahmad A (2018) Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers. Arab J Sci Eng 44(4): 3357–3368. https://doi.org/10.1007/s13369-018-3507-5

Witten I, Frank E, Hall M (2011) Data mining. Morgan Kaufmann, Burlington

Xiao L, Wan X, Lu X, Zhang Y, and Wu D (2018) IoT security techniques based on machine learning, arXiv preprint arXiv:1801.06275

Xue-qin Z, Chun-hua G, Jia-jun L (2006) Intrusion detection system based on feature selection and support vector machine. In: First international conference on communications and networking in China, Beijing, pp 1–5. https://doi.org/10.1109/CHINACOM.2006.344739

Zhang B, Yu Y, Li J (2018) Network intrusion detection based on stacked sparse autoencoder and BinaryTree ensemble method. In: Proceedings of the 2018 IEEE international conference on communications workshops (ICCWorkshops), Kansas city, MO, USA, 20–24 May 2018, pp 1–6

Zhang H, Yu X, Ren P, Luo C, Min G. (2019). Deep adversarial learning in intrusion detection: a data augmentation enhanced framework. arXiv 2019, arXiv:1901.07949

Zheng W (2020) Intrusion detection based on convolutional neural network. In: 2020 international conference on computer engineering and application (ICCEA), Guangzhou, China, pp 273–277. https://doi.org/10.1109/ICCEA50009.2020.00066

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.