Training a Quantum Annealing Based Restricted Boltzmann Machine on Cybersecurity Data

Vivek Dixit, Raja Selvarajan, Tamer Aldwairi, Yaroslav Koshka, Mark A. Novotny, Travis S. Humble, Muhammad A. Alam, *Fellow, IEEE*, and Sabre Kais

***Abstract*—A restricted Boltzmann machine (RBM) is a generative model that could be used in effectively balancing a cybersecurity dataset because the synthetic data a RBM generates follows the probability distribution of the training data. RBM training can be performed using contrastive divergence (CD) and quantum annealing (QA). QA-based RBM training is fundamentally different from CD and requires samples from a quantum computer. We present a real-world application that uses a quantum computer. Specifically, we train a RBM using QA for cybersecurity applications. The D-Wave 2000Q has been used to implement QA. RBMs are trained on the ISCX data, which is a benchmark dataset for cybersecurity. For comparison, RBMs are also trained using CD. CD is a commonly used method for RBM training. Our analysis of the ISCX data shows that the dataset is imbalanced. We present two different schemes to balance the training dataset before feeding it to a classifier. The first scheme is based on the undersampling of benign instances. The imbalanced training dataset is divided into five sub-datasets that are trained separately. A majority voting is then performed to get the result. Our results show the majority vote increases the classification accuracy up from 90.24% to 95.68%, in the case of CD. For the case of QA, the classification accuracy increases from 74.14% to 80.04%. In the second scheme, a RBM is used to generate synthetic data to balance the training dataset. We show that both QA and CD-trained RBM can be used to generate useful synthetic data. Balanced training data is used to evaluate several classifiers. Among the classifiers investigated, K-Nearest Neighbor (KNN) and Neural Network (NN) perform better than other classifiers. They both show an accuracy of 93%. Our results show a proof-of-concept that a QA-based RBM can be trained on a 64-bit binary dataset. The illustrative example suggests the possibility to migrate many practical classification problems to QA-based techniques. Further, we show that synthetic data generated from a RBM can be used to balance the original dataset.**

Manuscript received December 7, 2020; revised March 14, 2021; accepted April 8, 2021. *(Corresponding author: Vivek Dixit.)*

Vivek Dixit, Raja Selvarajan, and Sabre Kais are with the Department of Chemistry, Department of Physics and Astronomy, and Purdue Quantum Science and Engineering Institute, Purdue University, West Lafayette, IN 47907 USA (e-mail: dixitv@purdue.edu; selvarar@purdue.edu; kais@purdue.edu).

Tamer Aldwairi is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: taa70email@gmail.com).

Yaroslav Koshka is with the Department of Electrical and Computer Engineering, HPC2 Center for Computational Sciences, Mississippi State University, Mississippi State, MS 39762 USA (e-mail: ykoshka@ece.msstate.edu).

Mark A. Novotny is with the Department of Physics and Astronomy, HPC2 Center for Computational Sciences, Mississippi State University, Mississippi State, MS 39762 USA (e-mail: man40@msstate.edu).

Travis S. Humble is with the Quantum Computing Institute, Oak Ridge National Laboratory, Oak Ridge, TN USA (e-mail: humblets@ornl.gov).

Muhammad A. Alam is with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: alam@purdue.edu).

Digital Object Identifier 10.1109/TETCI.2021.3074916

***Index Terms*—RBM training, D-Wave quantum computer, quantum annealing, machine learning, synthetic data, cybersecurity, ISCX dataset.**

I. Introduction

NETWORKS have revolutionized our lives through various purposes including email, file transfer, web search, e-commerce, online banking, monetary transaction, education, collaboration, social networking, etc. The more we depend on it and use it, the more we expose ourselves to serious security risks. The internet is an insecure medium of communication. Any device connected to the internet is vulnerable. Cybersecurity is safety against cyber-attacks. Cyber-attacks are launched by hackers to gain unauthorized access or steal important data. The estimated total damage caused by global cybercrime has increased from $300 billion in 2013 to $945 billion in 2020 [1]. Financial loss from cybercrime is likely to increase in the coming years. Therefore, it is crucial to monitor dataflow in any network, and there is a need for robust software and devices that protect users from online security threats.

In this work, we investigate a restricted Boltzmann machine (RBM) coupled with quantum machine learning for a cybersecurity application. The application of quantum computing in machine learning is a promising technique, even with quantum computers currently being in an early stage of technological development. This paper is a first approach of implementing for network intrusion detection an analysis engine on a quantum computing device. A RBM is a generative model, which can be used to model the underlying probability distribution of a dataset. In addition to classifying data points, RBMs can also generate a new synthetic dataset. Despite the importance of the RBMs, only a few researchers have used RBMs for cybersecurity applications. Fiore *et al.* [2] used discriminative RBM for network anomaly detection applications. They showed that the performance of a model suffers when it is tested in a network different from the network that was used to obtain the training data. Aldwairi *et al.* [3] trained a RBM on the ISCX 2012 dataset using contrastive divergence (CD) and persistent contrastive divergence (PCD). Their model showed a percentage classification accuracy of 88.6% using CD and 89% for PCD. Alom *et al.* [4] applied a deep belief network (DBN) on the NSL-KDD D'99 intrusion detection dataset. They were able to get a classification accuracy of 97.5% just by using 40% of the dataset. A DBN model is composed of multiple layers of trained RBMs, weights are fine-tuned by performing backpropagation

in the final step of the model training. Salama *et al.* [5] used a DBN+SVM hybrid scheme for intrusion detection. They used a DBN for dimensionality reduction (from 41 to 5 features) and SVM for classification. The model was trained on the NSL-KDD dataset. Li *et al.* [6] trained a hybrid model on 10% KDDCUP'99 dataset. An autoencoder was used to reduce the dimensionality of the dataset and a DBN for classification. Alrawashdeh *et al.* [7] trained a DBN on the KDDCUP'99 dataset. Their model outperformed the model by Salama *et al.* [5] and Li *et al.* [6] both in speed and accuracy.

We have used a quantum annealer from D-Wave to train RBMs for intrusion detection applications and compared the performance to RBMs trained with contrastive divergence. Quantum annealers are based on adiabatic quantum annealing (QA), which is a powerful technique for optimization and sampling applications [8]–[12]. There are two main problems associated with the use of machine learning techniques for intrusion detection. The first problem is related to transferability and generalizability of the model, a model trained on a dataset performs poorly when tested on other datasets. The second problem is associated with the imbalanced nature of the cybersecurity dataset where the attack instances are outnumbered by benign instances, which makes detection of an attack like looking for a needle in a haystack. Quantum computing holds the promise to address these problems. A QA-trained RBM can effectively learn patterns without overfitting a dataset. Further, synthetic data from a RBM can be used to balance the original dataset. Our work is a step towards that goal. The D-Wave 2000Q adiabatic quantum computer has been used by several researchers for machine learning applications such as classification, regression, and clustering. Date *et al.* [13] used a quantum annealer for implementing linear regression. Willsch *et al.* [14] introduced a method to train support vector machines (SVMs) on a D-Wave 2000Q quantum annealer and compared its performance with classically trained SVMs. Kumar *et al.* [15] used quantum annealing to carry out the minimization of the clustering objective function. They implemented two clustering algorithms and compared their results with well-known k-mean clustering. Das *et al.* [16] used a D-Wave to implement a clustering algorithm for the clustering of charged particle tracks for a hadron collider experiment. Arthur *et al.* [17] used the D-Wave 2000Q adiabatic quantum computer to train the balanced k-means clustering model. They compared the results with classical k-means and classical balanced k-means. Kais *et al.* have used D-Wave's quantum annealer for prime factorization and electronic structure calculation of molecular systems [18], [19]. Adachi *et al.* [20] trained RBMs using a quantum annealer for a deep belief network (DBN) on a scaled-down MNIST dataset consisting of 32-bit length binary records. They showed that their model required fewer iterations than CD-based DBN training. Benedetti *et al.* [21] used quantum annealing to train a RBM on a 16-bit binary bars & stripes dataset. Koshka *et al.* [22], [23] trained a RBM using contrastive divergence and compared the samples obtained from Markov chain Monte Carlo (MCMC) and QA. For QA, the CD trained RBM was embedded onto the D-Wave, and sampling was performed. It was found that the QA based sampling revealed regions of the configuration space that were regularly missed by the MCMC based sampling, especially at medium to high energy (i.e., states of medium to low probability). Recently, Dixit *et al.* [24] trained a RBM using the D-Wave 2000Q quantum annealer for classification and image reconstruction applications. They used a 64-bit bars & stripes dataset in their work.

The D-Wave 2000Q has around 2000 qubits. D-Wave's recently introduced machine 'Advantage' comprises 5000 qubits. The number of qubits of a quantum annealer is a major factor that determines the size of a dataset that can be investigated. Sometimes the number of features of a large dataset can be reduced by finding a dense representation. Caldeira *et al.* [25] used PCA to reduce the number of features in the dataset. They used a QA-trained RBM for galaxy morphology classification. Sleeman *et al.* [26] used an autoencoder to obtain a dense reperesentation of their dataset. They were able to show nearly a 22-fold compression factor of grayscale 28 x 28 sized images to binary 6 x 6 sized images. They trained a QA-based RBM on the MNIST and the MNIST Fashion datasets.

II. Contribution

Cybersecurity is one of the key areas where the failure of detection systems can result in privacy intrusion, financial losses, and system shutdowns. Our goal is to train the RBM using a quantum annealer, to help explore quantum effects for faster training and to learn patterns efficiently. Given that network data is usually imbalanced, we seek to obtain synthetic samples generated by a RBM to provide rich information into the distribution from which attack samples are generated. This should enable classifiers to better train on and detect intrusions. There are two main objectives of this work. First, train a RBM using quantum annealing on a cybersecurity dataset (ISCX). Second, use a RBM to generate synthetic data to balance the cybersecurity data.

First, we show RBMs can be trained using a quantum annealer on a cybersecurity dataset. Conventional methods for RBM training such as CD and PCD are slow. They require many Gibbs cycles to train a RBM. Further, the CD does not estimate the correct gradient of log-likelihood [27]. RBM training using a quantum annealer is fundamentally different than existing methods. A quantum annealer exploits quantum effects like superposition and tunneling to find better low energy solutions. This could be particularly useful for intrusion detection applications where classifiers often show poor precision and accuracy. We believe that this is the first work that uses a QA-trained RBM for intrusion detection applications.

The second objective is to show that synthetic data from a RBM can be used to balance a cybersecurity dataset. Cybersecurity datasets often have a lower number of attack records. However, most of the machine learning techniques require a balanced dataset. A bias towards the majority class results if the dataset is not balanced. A commonly used method to balance a dataset is SMOTE (Synthetic Minority Over-sampling Technique) [28]. The SMOTE algorithm basically works by finding the k-nearest neighbor of a data point in the feature space of the minority class. Then a synthetic data point is obtained by interpolation between the data point and one of the k-neighbors.

Generally, this interpolation is performed based on a random number between 0 and 1. This process is repeated until the required number of synthetic data records is obtained. Several modifications and extensions of SMOTE have been made since its proposal [29]. Several investigators have used SMOTE for cybersecurity applications [30]–[34]. A trained RBM can be used to generate synthetic data records. An advantage of using a RBM is that the synthetic data from it follows the probability distribution of the training dataset. However, synthetic data from SMOTE might not follow the distribution of the training data. In this work, we use QA-trained as well as CD-trained RBMs to generate synthetic data. This synthetic dataset is then used to balance the original dataset.

Herein, we propose two schemes to balance the cybersecurity dataset. The first scheme is based on the under-sampling of benign records. In the second scheme, oversampling of the attack class is used. Synthetic data has been generated from a RBM to balance the training dataset. RBMs are trained on a benchmark intrusion detection dataset known as ISCX [35].

III. Methods

A. Restricted Boltzmann Machine (RBM)

A RBM is an undirected probabilistic graphical model consisting of a layer of visible variables and a single layer of latent or hidden variables. Each variable is connected to every variable in the opposite layer, but connections between the variables in the same layer are not allowed. Let the visible and hidden layers be composed of N and Mvariables denoted as $\{v_1, v_2, \ldots, v_N\}$ and $\{h_1, h_2, \ldots, h_M\}$, respectively. We collectively refer to the visible layer with the vector v and the hidden layer as h. The RBM is an energy-based model with the joint probability distribution specified by its energy function:

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)}, Z = \sum_v \sum_h e^{-E(v,h)}. \tag{1}$$

Z is the normalization constant known as the partition function. The energy function is defined as:

$$E(v,h) = -b^T v - c^T h - h^T W v, \tag{2}$$

where b and c are bias vectors at the visible and hidden layer, respectively; W is a weight matrix composed of w_{ij} elements.

B. Conditional Distribution

The probability of getting a vector h at the hidden layer given a vector v at the visible layer is:

$$P(h|v) = \frac{P(v,h)}{P(v)} \tag{3}$$

where $P(v)$ is given by the following expression:

$$P(v) = \frac{\sum_h e^{-E(v,h)}}{Z}. \tag{4}$$

Using expression $P(v,h)$ from 1, we get:

$$P(h|v) = \frac{\exp\{\sum_j c_j h_j + \sum_j (v^T W)_j h_j\}}{Z'}, \tag{5}$$

where

$$Z' = \sum_h \exp(c^T h + h^T W v). \tag{6}$$

$$P(h|v) = \frac{1}{Z'} \prod_j \exp\{c_j h_j + (v^T W)_j h_j\}. \tag{7}$$

Let's denote

$$\widetilde{P}(h_j|v) = \exp\{c_j h_j + (v^T W)_j h_j\} \tag{8}$$

The probability to find an individual variable in the hidden layer, $h_j = 1$ is:

$$\begin{aligned} P(h_j = 1|v) &= \frac{\widetilde{P}(h_j = 1|v)}{\widetilde{P}(h_j = 0|v) + \widetilde{P}(h_j = 1|v)} \\ &= \frac{\exp\{c_j + (v^T W)_j\}}{1 + \exp\{c_j + (v^T W)_j\}} \end{aligned} \tag{9}$$

Thus, the individual hidden activation probability is given by:

$$P(h_j = 1|v) = \sigma\left(c_j + (v^T W)_j\right), \tag{10}$$

where σ is the logistic function. Similarly, the activation probability of a visible variable conditioned on a hidden vector h is given by:

$$P(v_i = 1|h) = \sigma\left(b_i + (h^T W)_i\right). \tag{11}$$

C. RBM Training

A RBM is trained by maximizing the likelihood of the training data. The log-likelihood is given by:

$$\begin{aligned} l(W,b,c) &= \sum_{t=1}^{N} \log P\left(v^{(t)}\right) \\ &= \sum_{t=1}^{N} \log \sum_h P\left(v^{(t)}, h\right), \end{aligned} \tag{12}$$

where N is the number of records in the training dataset and $v^{(t)}$ is a sample from the training dataset.

$$\begin{aligned} l(W,b,c) = &\sum_{t=1}^{N} \log \sum_h e^{-E(v^{(t)},h)} \\ &- N \cdot \log \sum_{v,h} e^{-E(v,h)}. \end{aligned} \tag{13}$$

Denote $\theta = \{W, b, c\}$. The gradient of the log-likelihood is given by:

$$\begin{aligned} \nabla_\theta l(\theta) = &\sum_{t=1}^{N} \frac{\sum_h e^{-E(v^{(t)},h)} \nabla_\theta(-E(v^{(t)},h))}{\sum_h e^{-E(v^{(t)},h)}} \\ &- N \cdot \frac{\sum_{v,h} e^{-E(v,h)} \nabla_\theta(-E(v,h))}{\sum_{v,h} e^{-E(v,h)}} \end{aligned} \tag{14}$$

$$\begin{aligned} \nabla_\theta l(\theta) = &\sum_{t=1}^{N} \langle \nabla_\theta(-E(v^{(t)},h)) \rangle_{P(h|v^{(t)})} \\ &- N \cdot \langle \nabla_\theta(-E(v,h)) \rangle_{P(v,h)}, \end{aligned} \tag{15}$$

where $\langle\cdot\rangle_{P(v,h)}$ is the expectation value with respect to the distribution $P(v,h)$. The gradient with respect to θ can also be expressed in terms of its components:

$$\nabla_w l = \frac{1}{N}\sum_{t=1}^{N}\langle v^{(t)}\cdot h\rangle_{P(h|v^{(t)})} - \langle v\cdot h\rangle_{P(v,h)} \tag{16}$$

$$\nabla_b l = \frac{1}{N}\sum_{t=1}^{N}\langle v^{(t)}\rangle_{P(h|v^{(t)})} - \langle v\rangle_{P(v,h)} \tag{17}$$

$$\nabla_c l = \frac{1}{N}\sum_{t=1}^{N}\langle h\rangle_{P(h|v^{(t)})} - \langle h\rangle_{P(v,h)}. \tag{18}$$

The first term in 15 is a data-dependent term. It can be exactly calculated using a training vector $v^{(t)}$ and a hidden vector h. Given $v^{(t)}$, the vector h can be calculated using 10. The second term is a model-dependent term. Getting samples for the second term is difficult. The Contrastive Divergence (CD) is the most commonly used algorithm to determine the model-dependent term. In CD, a training vector is applied to the visible layer. Then the binary states of the hidden units are computed in parallel using 10. The states of the units on the visible layer are reconstructed using h via 11. Finally, the reconstructed v is used to find a new h on the hidden layer. During the RBM training, the change in model parameters is given as:

$$\theta_j^{new} = \theta_j^{old} + \epsilon\cdot\nabla_{\theta_j} l(\theta_j) \tag{19}$$

where ϵ is the learning rate.

The learning works well even though CD only crudely approximates the gradient of the log probability of the training data. Sutskever *et al.* [36] have shown that the contrastive divergence does not estimate the gradient of the log-likelihood. An effective method for RBM training is still not known. It has been shown by several researchers that a RBM can be trained using samples drawn from the D-Wave quantum annealer [20], [21], [25], [24]. The first term of the gradient of the log-likelihood is estimated using the procedure explained earlier. The second term which is the model-dependent term is calculated in the following way. First, a RBM is embedded on to a quantum annealer, then quantum annealing is performed. The samples obtained from quantum annealing are used to compute the second term. The samples from a quantum annealer operating at a temperature T is qualitatively similar to a probability distribution given by $\exp(\frac{-E(v,h)}{kT})$. However, to compute the model-dependent term we need samples from a distribution $\exp(-E(v,h))$ (15). To address this problem we scale the energy by a hyperparameter S, such that for the model-dependent term, we sample from the $\exp(\frac{-E(v,h)}{SkT})$ distribution. Here, S is a hyperparameter, which is determined by a manual search. The optimal condition corresponds to the case when $SkT = 1$. We keep S fixed during the entire RBM training process. However, the temperature T generally changes. This mismatch between S and T might result in suboptimal RBM training. An efficient way to compute T at each training step is still not discovered. It should be noted as the RBM training starts with random weights and biases, samples from the D-Wave are not expected to show a Boltzmann distribution, however, as the training progresses the underlying probability distribution moves toward the Boltzmann distribution. RBM training using CD-1 and QA is summarized in Algorithm 1 and Algorithm 2, respectively. The two methods differ only in the manner the model-dependent term is estimated.

D. The D-Wave Quantum Annealer

To formulate a problem for the D-Wave, one needs to transform the problem into Ising form given by:

$$E(s|h,J) = \sum_{i=1}^{N} h_i s_i + \sum_{i<j}^{N} J_{ij} s_i s_j;\ s_i \in \{-1,+1\}. \tag{20}$$

This is an objective function of N variables $s = [s_1, s_2, \ldots, s_N]$ corresponding to physical Ising spins, where h_i are the biases and J_{ij} the couplings between spins.

The energy of a RBM model given by 2, has a form similar to 20. The weights and biases of a RBM which is trained using a binary dataset, $\{0,1\}$ states can be converted to use $\{-1,1\}$ states via the mapping [37]:

$$b_i' = \frac{b_i}{2} + \frac{\sum_j W_{ij}}{4} \tag{21}$$

$$c_i' = \frac{c_i}{2} + \frac{\sum_j W_{ij}}{4} \tag{22}$$

$$W' = \frac{W}{4}. \tag{23}$$

These weights and biases can be used to embed a RBM onto the D-Wave machine. After executing quantum annealing, solutions can be sampled. We set the anneal time to 20 μs for each anneal. The resulting bipolar samples may be converted to a binary sample simply by replacing all instances of -1 with 0.

The D-Wave 2000Q quantum annealer has 2048 qubits arranged in 16×16 unit cells forming a C16 Chimera graph. Each unit cell is composed of 8 qubits connected in a bipartite graph. Each qubit is connected to four other qubits of the same unit cell and two qubits of two different unit cells. One can embed a fully connected RBM of 64 visible and 64 hidden units on a C16 Chimera graph as shown in Fig. 1. In this embedding, each RBM unit is represented by a chain of 16 physical qubits. If we look at the arrangement of qubits, we note that 16 qubits can be combined by forming a vertical chain. Each vertical chain forms one visible unit. Similarly, 16 horizontal qubits can be linked together to form a hidden unit. In Fig. 1, the vertical chains are shown in red, while the horizontal chains are in black. There are 64 vertical and 64 horizontal chains which represent 64 visible units and 64 hidden units of the RBM. If some of the qubits in the D-Wave QPU are missing or not working, then the length of the qubit chain forming a RBM unit will be shorter. In that case, the RBM will be not fully connected, that is some connections between visible and hidden units will be missing. Fortunately, in the D-Wave QPU, only a few qubits are missing which does not seem to affect the performance of the RBMs. This embedding has also been used in our previous work [24]. A similar bipartite embedding has been demonstrated by other researchers [38]–[40]. For the lattice with almost no missing qubits and couplings, this embedding is close to optimal for a Chimera graph. We have maximally made use of the D-Wave 2000Q to allow for 64 hidden qubits and visible qubits that

Fig. 1. Minor-Embedding a RBM with 64 visible and 64 hidden units on the D-Wave 2000Q. Each visible (hidden) unit is made by forming a vertical (horizontal) chain of 16 physical qubits shown in red (black).

Algorithm 1: Optimization of Learning Parameters Using CD-1

$\epsilon \leftarrow learningrate$ ▷ϵ, is the step size, a small positive number.
$b, c, W \leftarrow randomnumber$ ▷Initialize with small normally distributed random numbers.
while *not converged* **do**
  Sample a minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from the training set
  $V \leftarrow \{x^{(1)}, \ldots, x^{(m)}\}$
  $H \leftarrow \sigma(c + VW)$ ▷σ is the logistic function
  $V' \leftarrow \sigma(b + HW^T)$
  $H' \leftarrow \sigma(c + V'W)$
  $W \leftarrow W + \epsilon\frac{(VH - V'H')}{m}$ ▷updates W
  $b \leftarrow b + \epsilon\frac{(sum(V) - sum(V'))}{m}$ ▷updates b
  $c \leftarrow c + \epsilon\frac{(sum(H) - sum(H'))}{m}$ ▷updates c
end

are two way fully connected to each other. Using any other embedding would result in a small size of the feature space and hence is not preferred. The newer machine (Advantage) has over 5000 qubits and additional graph connections, which should allow for an extended feature space size where one could do a comparison of the performance of different embedding schemes for this dataset.

E. Evaluation Metrics

To compare and quantify the performance of different methods, metrics based on a confusion matrix are used. For a binary classification problem that has two classes namely 'positive' and 'negative' important metrics for model evaluation are:

$$A_{tot} = \frac{TP + TN}{TP + TN + FP + FN} \times 100, \tag{24}$$

$$A_P = \frac{TP}{TP + FP} \times 100, \tag{25}$$

$$Precision = \frac{TP}{TP + FP} \tag{26}$$

$$Recall = \frac{TP}{TP + FN} \tag{27}$$

$$F_1 score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{28}$$

where TP (true positive) and FP (false positive) are the number of correctly and incorrectly predicted observations of class 'positive,' respectively. Similarly, TN (true negative) and FN (false negative) are the number of correctly and incorrectly predicted observations of class 'negative,' respectively. A_{tot} is the total percentage of classification accuracy; A_P is the percentage of classification accuracy of the class 'positive'. Precision is the ability of the model not to predict the label of a sample of a class incorrectly, while recall is the ability of the model to correctly predict all the samples of a class correctly. The F_1 score is the harmonic mean of precision and recall. A robust classifier will have a high value of the F_1 score. Precision, recall, F_1 score, and percentage accuracy are used as metrics to evaluate models.

Algorithm 2: Optimization of Learning Parameters Using Quantum Annealing.

$\epsilon \leftarrow learningrate$ ▷ϵ, is the step size, a small positive number.
$b, c, W \leftarrow randomnumber$ ▷Initialize with small normally distributed random numbers.
while *not converged* **do**
  Sample a minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from the training set
  $V \leftarrow \{x^{(1)}, \ldots, x^{(m)}\}$
  $H \leftarrow \sigma(c + VW)$ ▷σ is the logistic function
  $\{h, J\} \leftarrow \{b, c, W\}$
  $(V', H') \leftarrow quantumannealing(h, J, S)$
  $W \leftarrow W + \epsilon\frac{(VH - V'H')}{m}$ ▷updates W
  $b \leftarrow b + \epsilon\frac{(sum(V) - sum(V'))}{m}$ ▷updates b
  $c \leftarrow c + \epsilon\frac{(sum(H) - sum(H'))}{m}$ ▷updates c
end

F. Material Setup

The D-Wave 2000Q quantum annealer has been used to obtain samples of training QA-based RBM. The D-Wave operates at a temperature that is fixed based on the training results. The temperature corresponds to an effective scaling of parameters that are supplied as coupling weights and biases to the machine.

For training CD-based RBM, a personal computer has been used. In-house codes were developed to implement RBM training using CD and to obtain samples from the quantum annealer.

MATLAB programming language is used. MATLAB codes are also developed for classification and to generate synthetic data. To implement popular classification methods namely: Neural Networks, K-nearest neighbor, Support Vector Machine, Decision tree, and Naive Bayes, machine learning library for python programming language 'scikit-learn' [41] has been used. All the classifiers are trained on 62-bit binary input and 2-bit output data. A neural network with five layers has been used. Layer 1 has 62 nodes. There are three hidden layers, each with 8 nodes. The output layer has 2 nodes. The 'relu' activation function and 'adam' solver have been used for training. To implement the K-nearest neighbor classifier neighbors is set 3, other parameters are set to their default values. The decision tree classifier has a max depth set to 5, other values are set to the default. To implement support vector machine and Naive Bayes classifiers, SVC and GaussianNB classifiers of 'scikit-learn' have been used. For these two classifiers, all the parameters are set to their default values.

G. Dataset

This study investigates a cybersecurity benchmark dataset known as 'ISCX IDS dataset 2012'. We will call it ISCX, for brevity. The ISCX is one of the publicly available datasets on the website of the Canadian Institute of Cybersecurity at https://www.unb.ca/cic/datasets/index.html. The ISCX consists of seven days of network activity. There are two main classes namely 'benign' and 'attack'. For more details and the underlying approach that is used to generate this dataset, see [35]. The preprocessing of the data consists of the conversion of variables from categorical to numerical, dimensionality reduction, and binarization of numerical data. All these steps are necessary to build a classifier based on a QA trained RBM. Finally, the dataset is binarized by using a supervised discretization filter implemented in Weka [42]. As we discussed earlier, one can embed a RBM onto the D-Wave 2000Q with 64 visible and 64 hidden units. Therefore, we set the total number of columns in the binarized dataset to be 64. There are 62 binary features in the dataset and the last two columns are the target variables. When the last two bits are 01, it indicates a benign instance; while 10 indicates an attack. If the last two bits are either a 00 or 11, it indicate an indeterminate case. Thus, the possibility that a random guess could be correct is 25%, and keeping two bits for the target variable helps to prepare a more robust machine learning model as compared to the case where one bit is used as a target variable.

IV. Results

The dataset that was obtained after binarization of the original dataset had 137 584 instances. However, it was found that most of the records were repeated. The dataset was further modified, and only unique records were retained. There were 25 230 unique benign and 4917 unique attack records. Training and test datasets are formed with these records. The test dataset comprises 500 attack and 500 benign records. The remaining 29 147 unique records are used in the training dataset. We trained a RBM on the training dataset. The classification accuracies for the attack and benign classes are found to be 42% and 97%, respectively. These accuracies are estimated on the test dataset. The lower accuracy for the attack class could be attributed to the fact that there are a significantly higher number of benign instances in the dataset compared to the number of attack instances. The attack records constitute only 14.1% of the total dataset, however, ideally, there should be 50% records of each class. The problem of an imbalanced dataset is commonly seen in cybersecurity datasets; attack records form a rarer class. Machine learning algorithms show the best results when the number of observations in each class is almost similar. Thus, an imbalanced dataset leads to a poor classification performance of the model. This imbalanced dataset is also investigated using other classification methods, and the results are presented in Table III. To tackle the problem of an imbalanced dataset we propose two schemes. In the first scheme, we use undersampling of the benign class, while in the second scheme a RBM has been used to generate instances in order to balance the training dataset. These schemes are discussed in detail in the following sections.

A. Scheme 1: Balancing Training Data by Undersampling of Benign Records

Scheme 1 is illustrated in Fig. 2. In this approach, the binarized dataset is divided into training and testing datasets. The training dataset is composed of 21 450 records (Benign=18 000, Attack=3450), while the test data contains 8697 records (Benign=7230, Attack=1467). Thus, the original binarized dataset is divided into training and testing datasets in a ratio of $\approx 70\% : 30\%$. The training dataset is further divided into five smaller datasets namely A, B, C, D, and E. The total number of benign records in the training dataset is divided into five datasets as $18000 = 3450 + 3450 + 3450 + 3450 + 4200$. Thus, each sub-dataset has unique benign records. There are 3450 attack records in the training dataset, we add the same 3450 attack records to each sub-datasets. Sub-dataset E has 4200 ($=3450 + 750$) attack records, 750 of which are repeated. Thus, each sub-dataset contains an equal number of instances of both classes. Five RBM models are trained on these five datasets. These trained RBMs models are used to make predictions on the testing dataset. Predictions from the five RBM models are collected and a majority vote rule has been performed to obtain a final result. Two different methods, contrastive divergence (CD-1) and quantum annealing (QA), are employed to train the RBMs. In Table I we show the average classification accuracy of the benign class is 90.51% and that of the attack class is 88.94%. The total accuracy is 90.24%. On using the majority vote on the results obtained from five different RBMs, the classification accuracies with which the benign and attack classes can be predicted, and total accuracy have been found to be 96.17%, 93.25%, and 95.68%, respectively. In the case where RBMs are trained with quantum annealing, the average classification accuracy of the benign and attack classes, and total accuracy are 73.62% and 71.18%, and 74.14%, respectively. On applying the majority vote on the results from five trained RBMs, the average classification accuracy for the benign, attack classes, and total accuracy are found to be 74.46%, 85.62%, and

TABLE I
THE ORIGINAL TRAINING DATASET IS DIVIDED INTO FIVE SMALL DATASETS (A, B, C, D, AND E). THESE DATASETS ARE USED TO TRAIN FIVE RBMS USING CONTRASTIVE DIVERGENCE (CD-1) AND QUANTUM ANNEALING (QA). CLASSIFICATION ACCURACIES FOR BENIGN AND ATTACK CLASSES, AS WELL AS TOTAL ACCURACY, ARE PRESENTED FOR EACH DATASET. VALUES ARE EVALUATED ON THE TESTING DATA. A MAJORITY VOTE IS PERFORMED ON THE RESULTS OBTAINED FROM FIVE RBMS

dataset	No. of records	Accuracy, A_{benign}		Accuracy, A_{attack}		Total Accuracy, A_{tot}	
		CD-1 (%)	QA (%)	CD-1 (%)	QA (%)	CD-1 (%)	QA (%)
A	6900	92.17	82.60	88.75	68.30	91.59	80.19
B	6900	88.30	71.76	88.82	72.60	88.39	71.90
C	6900	90.11	69.04	89.23	83.01	89.96	76.03
D	6900	90.03	67.65	90.52	72.73	90.11	68.51
E	8400	91.94	77.05	87.39	59.24	91.17	74.05
Average	–	90.51	73.62	88.94	71.18	90.24	74.14
Standard deviation	–	1.59	6.17	1.12	8.59	1.25	4.38
Majority Vote	–	96.17	74.46	93.25	85.62	95.68	80.04

TABLE II
SYNTHETIC DATASETS ARE GENERATED FROM RBMS TRAINED USING CONTRASTIVE DIVERGENCE (CD) AND QUANTUM ANNEALING (QA). THESE SYNTHETIC DATASETS ARE THEN USED TO TRAIN RBMS USING CD. THE CLASSIFICATION ACCURACIES OF THESE RBMS FOR BENIGN AND ATTACK RECORDS, AS WELL AS TOTAL ACCURACIES, ARE PRESENTED. THESE ACCURACIES ARE CALCULATED ON THE TEST DATASET. THE LABEL 'MODEL' INDICATES THE RBM MODEL THAT WAS USED TO GENERATE THE SYNTHETIC DATASET

Model	Classification Accuracy			Model	Classification Accuracy		
	Benign (%)	Attack (%)	Total (%)		Benign (%)	Attack (%)	Total (%)
A-CD	78.99	80.91	79.31	A-QA	63.35	76.89	65.63
B-CD	73.26	84.32	75.13	B-QA	67.34	73.62	68.40
C-CD	59.35	90.66	64.63	C-QA	72.28	66.73	79.42
D-CD	61.69	87.12	65.98	D-QA	72.89	69.05	72.24
E-CD	68.13	85.28	71.02	E-QA	61.51	78.19	64.32
Average	68.28	85.66	71.21	Average	67.47	72.90	70.00
Stdev	8.10	3.59	6.16	Stdev	5.12	4.93	6.08

TABLE III
BALANCED TRAINING DATA IS USED TO TRAIN SIX CLASSIFIERS. PERFORMANCE METRICSES: PRECISION, RECALL, F_1 SCORE, AND ACCURACY ARE USED TO COMPARE MODELS. THE LABEL 'CD-BAL' ('QA-BAL') INDICATES THAT THE SYNTHETIC DATA THAT IS USED TO BALANCE THE TRAINING DATASET IS OBTAINED FROM A RBM TRAINED WITH CONTRASTIVE DIVERGENCE (QUANTUM ANNEALING). THE LABEL 'IMBAL' INDICATES THE ORIGINAL IMBALANCED DATASET. VALUES ARE EVALUATED ON THE TESTING DATA

Method	Data	Precision		Recall		F_1 score		Accuracy, A_{tot}
		Attack	Benign	Attack	Benign	Attack	Benign	
Restricted Boltzmann Machine	CD-bal	0.87	0.94	0.95	0.85	0.91	0.89	85%
	QA-bal	0.92	0.82	0.80	0.93	0.85	0.87	82%
	imbal	0.93	0.63	0.42	0.97	0.58	0.77	68%
Neural Network (NN)	CD-bal	0.95	0.90	0.89	0.96	0.92	0.93	93%
	QA-bal	0.95	0.90	0.89	0.96	0.92	0.93	93%
	imbal	1.00	0.60	0.32	1.00	0.49	0.75	66%
K-Nearest Neighbor (KNN)	CD-bal	0.98	0.89	0.87	0.98	0.92	0.93	93%
	QA-bal	0.98	0.89	0.87	0.98	0.92	0.93	93%
	imbal	1.00	0.62	0.40	1.00	0.57	0.77	70%
Support Vector Classifier	CD-bal	0.94	0.77	0.70	0.95	0.80	0.86	83%
	QA-bal	0.94	0.77	0.70	0.95	0.80	0.85	83%
	imbal	1.00	0.58	0.26	1.00	0.42	0.73	63%
Decision Tree	CD-bal	0.90	0.83	0.80	0.92	0.85	0.87	86%
	QA-bal	0.90	0.83	0.80	0.92	0.85	0.87	86%
	imbal	1.00	0.56	0.22	1.00	0.36	0.72	61%
Naive Bayes	CD-bal	0.87	0.60	0.33	0.95	0.48	0.74	65%
	QA-bal	0.87	0.60	0.33	0.95	0.48	0.74	65%
	imbal	1.00	0.54	0.16	1.00	0.27	0.70	58%

80.04%, respectively. Thus, in the case of CD-1 as well as QA, we note an improvement in accuracy when the majority vote is applied. Table I also compares the performances of RBMs trained using CD-1 and QA methods. Using the majority vote the total accuracy with CD-1 and QA methods are found to be 95.68% and 80.04%, respectively. If we consider the results from the individual models, for example, the RBM model trained on sub-dataset A. Dataset A is comprises of just 3450 attack and 3450 benign records, but the classification accuracy of the RBM is better than the case when the training dataset was imbalanced (Table III). The contrastive divergence being a state-of-the-art method for RBM training, a better performance of a CD trained

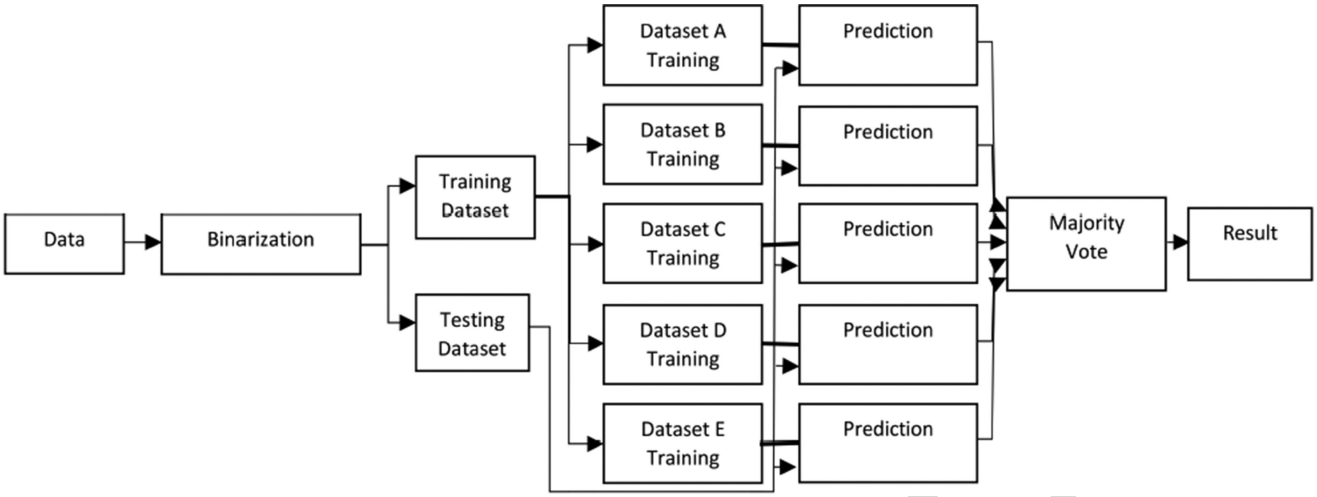


Fig. 2. Scheme 1: Flowchart for intrusion detection using an imbalanced dataset. The training dataset is divided into five balanced datasets which are subsequently used to train five classifiers. The final classification result is obtained by a majority voting.

RBM is expected. While CD-1 is a popular and effective method for RBM training, QA for RBM training has not been substantially explored. Considering the prevailing noise and error-prone nature of the existing quantum machine a classification accuracy of 80.04% seems to be satisfactory. Our goal here is to show a proof-of-concept that RBM can be trained using quantum annealing on a 64-bit binary dataset. RBM training using QA can be improved by optimizing D-Wave annealing parameters like anneal time, chain length, etc. Further, an efficient way to calculate the quantum annealer's effective temperature can also improve QA-based RBM training.

B. Scheme 2: Balancing Training Dataset With Synthetic Data

A dataset is said to be imbalanced if the number of observations in each class is not proportionate. Generally, when we deal with a cybersecurity dataset, we face the problem of a lower number of attack instances compare to the benign instances. Previously, we showed this problem could be solved by creating several small sub-datasets and subsequently using those to train individual models, and finally reaching a result by performing a majority vote. Another way to deal with this problem is to generate synthetic data using a RBM and then using the synthetic data to balance the training dataset.

In this section, we will discuss how synthetic data generated from a trained RBM has been used to balance the training dataset. A synthetic data sample can be generated from a RBM trained using CD-1 in the following way. We input a 64-bit vector formed using random 0 s and 1 s to a trained RBM. After 50 Gibbs cycles, we sample a 64-bit binary vector from the visible layer of the RBM. This sampled binary vector forms an instance of the synthetic dataset. Generating a synthetic dataset using QA is straightforward. One needs to embed a trained RBM onto the D-Wave quantum annealer and perform a quantum annealing step. For quantum annealing the anneal time was set to 20 μs for each anneal and the number of samples that were requested was 10 000. Thus, 10 000 samples can be obtained from the quantum annealer very quickly (1000 results within tens of milliseconds). From each sample, the states of the visible units are determined. Each sample corresponds to a record in the synthetic dataset. In this way, synthetic data composed of 10 000 records is obtained using QA.

To ensure that the synthetic dataset generated from a trained RBM is useful, we perform the following experiment. We use trained RBM models (A, B, C, D, and E) from the previous experiment to generate synthetic datasets; one from each model. Thus, ten datasets are generated; five from the CD-based RBMs and the other five from the QA-based RBMs. Now, ten RBM models are trained on these ten synthetic datasets using CD-1. The performance of these ten RBM models is compared by estimating classification accuracies on the test dataset composed of 8697 records (Benign=7230, Attack=1467). The results from these RBM models as well as estimated average and standard deviation are presented in Table II. RBMs trained using synthetic dataset generated from 'CD-1 trained RBM' shows total classification accuracy varying between 64.63% to 79.31%. The RBM trained with synthetic dataset obtained from 'QA trained RBM' shows classification accuracy varying between 64.32% to 79.42%. The average classification accuracy for benign and attack classes are 68.28% and 85.66% with dataset obtained from CD-1 based RBM, and 67.47% and 72.90% with dataset obtained from QA based RBM. The results from Table II indicate that useful synthetic data can be generated from a trained RBM. This synthetic data can be used to augment the original imbalanced training dataset in order to balance it. Also, on the basis of the classification accuracies, one can conclude that the samples obtained from a RBM trained using QA are as good as from a RBM trained with CD-1.

Now we know that a RBM can be used to generate useful synthetic data. We can use this procedure to generate synthetic data to balance the training data and hence improve the performance of a classifier. Scheme 2, which uses a RBM to generate new data, is illustrated in Fig. 3. The original dataset is first binarized and divided into testing and training data. The training data is used to

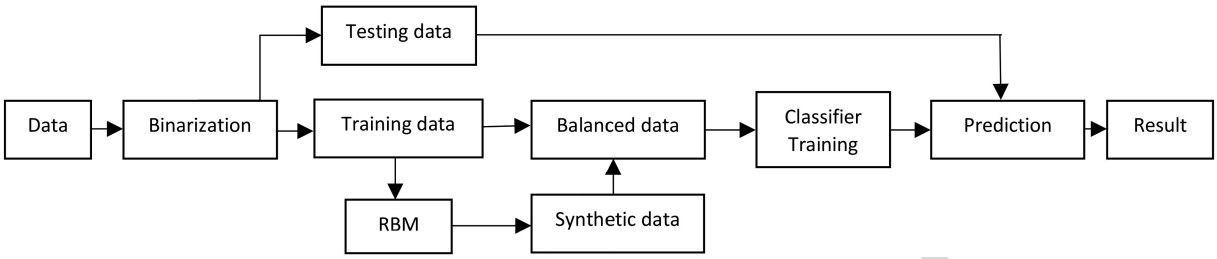


Fig. 3. Scheme 2: Flowchart for intrusion detection using an imbalanced dataset. A RBM is first trained using the training dataset and then it is used to generate synthetic data. Training data and synthetic data are used to create a balanced dataset which is further used to train a classifier.

train a RBM, which is subsequently used to generate a synthetic dataset. Depending on the number of instances needed to balance the training data, one can use a subset of a synthetic dataset to balance the training dataset. There were 18 000 benign and 3450 attack records in the training dataset, so 14 550 synthetic attack records are added to balance the training dataset. A classifier is then trained on the balanced dataset and a prediction on the original testing data is performed.

Considering the fact that RBMs are mostly used as a generative model and there are other classification methods that perform better than a RBM classifier, we train several classifiers on the balanced training dataset. The results are presented in Table III. For comparison, model performance with the original imbalanced dataset is also included. We notice in the table that K Nearest Neighbor (KNN) and Neural Network (NN) performed better than other models. They both showed a classification accuracy of 93%. Their values for precision, recall, and F_1 scores are also better than other methods. The lowest value of classification accuracy, as well as other metrics, are found in the case of Naive Bayes. The classification accuracy for this classifier is 65%. This exercise shows that it is important to investigate different classifiers to achieve better performance and different methods may give widely differing results. Table III shows that all classification methods show improved performance when the dataset is balanced. Thus, the RBM-based technique that is used to balance the dataset using synthetic data is effective. This demonstrates the ability of the RBM to fill gaps in an imbalanced dataset by creating synthetic data that falls within the probability range of existing data.

V. Discussion

Quantum computers are still in a formative stage of their technology. Consequently, comparing the RBM approach using QA to the mature classical CD or other approaches is uneven. The QA approach is expected to progress as quantum computing technology advances. In scheme 1, we note that the total classification accuracies using QA-trained and CD-trained RBM are 80.04% and 95.68%, respectively. The performance gap that arises between CD-trained and QA-trained could be attributed to the following reasons. First, it has been observed by Koshka *et al.* [22], [23] that RBM sampling using QA misses many of the higher-energy regions of the configuration space, while also finding many new regions consistently missed by CD. Perhaps in the present case of the ISCX dataset, high energy samples missed by QA are also important. The overall effect is the RBM learns, but not as well as we expect. Another reason could be an instance-dependent effective temperature of the D-Wave annealer. We would like the D-Wave to sample with $kT = 1$, where T refers to the temperature at which the D-Wave operates. However, this is hardly the case and hence we introduce an effective scaling parameter S, for the Hamiltonian being embedded that allows us to ensure SkT approximates unity. The effective scaling is treated as a hyperparameter and is fixed throughout the training of the RBM. Ideally one should calculate an effective temperature during each training epoch. This mismatch might degrade RBM's learning during the training. An accurate way to estimate the temperature at which the D-Wave samples for ground-state configuration is an open challenge. Efforts have been made towards identifying instance-dependent temperature for smaller models, none of which have proven to scale efficiently towards larger feature spaces [21], [25]. Finally, hardware limitations like limited connectivity (which forces one to form long chains), quantum noise, low coherence time, etc could be some other reasons for the lower classification performance of the QA-based approach.

When we compare synthetic data obtained from QA-trained RBM and CD-trained RBM (Table II), we do not see much difference in classification performances. Accuracies of RBMs trained on both datasets are similar. These results indicate that our simplified approach of using a hyperparameter instead of an exact instance-dependent temperature is useful. There is another advantage of using QA for RBM training. Depending on the complexity of a dataset, the CD might need hundreds of Gibbs cycles to reach the equilibrium to finally give one sample, while using a QA-based approach one can obtain 10 000 samples almost instantaneously. Further, with the availability of quantum annealers with higher qubits and better connectivity, lower noise, the QA-based RBM training is likely to be improved and it would be possible to deal with larger datasets. Several investigators have shown that by employing machine learning techniques like principal component analysis and autoencoders to compress data, one can investigate a moderate size dataset with currently available quantum annealers [25], [26]. CD-based and QA-based approaches are fundamentally different ways of training RBMs. It would be an interesting exercise to train a RBM using samples obtained from both methods together. After training one should compare the results with RBMs trained

separately using QA and CD approaches. Our results indicate tha a RBM could be an effective tool to generate synthetic data that can be used to balance a dataset. One could also try training the RBM exclusively on the minority class to balance the original dataset. The QA-based approach can be used for faster sampling as sampling from a quantum annealer is almost instantaneous.

We see that it is much easier to compute the model term using a D-Wave to sample low energy eigenstates of the Hamiltonians. This shows how a D-Wave machine can be utilized in problems beyond optimization and into the machine learning world. However, there are some limitations of the quantum annealer that we use. The D-Wave 2000Q allows for a fully connected bilayer network of only 64 qubits at maximum. This limits the size of feature space of the data to be used for doing a study on large datasets without using another layer of feature extraction to downsize the data set used for the study. The qubits are noisy and less coherent compared to the upcoming new D-Wave machine which supports 5000 qubits and additional qubit interconnectivity and this provides opportunities for doing better analysis of the proposed schemes. A larger feature space would also allow for more confident claims to be made about the role that these machines might play in the machine learning world.

There are several advantages of using the D-Wave quantum annealer for RBM training. It offers a fundamentally different way to compute the model dependent term of the gradient of log-likelihood. Computation of this term using conventional methods like CD and PCD is intractable. Further, QA based sampling is faster than MCMC used in CD or PCD. So, we expect that with improvement in hardware such as more qubits, lower noise, better coherence time as well as robust algorithm for effective temperature, the QA-based RBM training is likely to perfom better than the CD-based approach.

The dataset that we use in this study is imbalanced. It comprises 30 147 unique records. The number of records that belong to the attack class is 4917. It looks like the amount of data of the attack class is not enough for RBM training. However, when we balance the dataset and train classifiers on it, the results indicate that the data amount is sufficient. For example, in the case of RBM trained on the "CD-bal" dataset, the precision, recall, and F_1 score for the attack class are 0.87, 0.95, and 0.91, respectively (Table III). It seems like though the number of attack records and features is small, the chosen records/features are representative of the model. Aldwairi *et al.* [43] established that when certain features that are representative of the model are to be selected, the change in the accuracy is minimal across all tested algorithms. Our first approach which uses under-sampling of benign records as well as the second approach where oversampling of attack records is used, seem to be effective for balancing the ISCX dataset.

VI. Conclusion

Restricted Boltzmann machine (RBM) methodology has been investigated for classification and synthetic data generation using the cybersecurity ISCX dataset. RBMs are trained through a quantum annealing approach performed using the D-Wave 2000Q quantum annealer. For comparison, a state-of-the-art method for RBM training, contrastive divergence, is also investigated. The ISCX dataset is preprocessed and binarized to transform it into a form that can be used with a RBM. When a classifier is trained on the original data, it is found that attack records can be correctly predicted with an accuracy of 42%, while benign records are predicted with an accuracy of 97%. This disproportionate result is attributed to the fact that the dataset is imbalanced. The attack records in the dataset only account for 14.1% of the total number of records. To deal with the imbalanced dataset, we propose two schemes. The first scheme is based on the undersampling of benign records. In this scheme, the training dataset is divided into five sub-datasets. Five classifiers are trained separately on these datasets. The final result has been obtained by performing a majority voting on the results from the individual classifiers. Our results show that by using a majority vote the classification accuracy increased up to 95.68% from 90.24% in the case of CD-1. In the case of QA, the classification accuracy increased to 80.04% from 74.14%. The second scheme that we use to balance the training dataset is based on the generation of synthetic data using a trained RBM. The balanced dataset obtained from this scheme is used to train six different classifiers. Neural network and K-nearest neighbor models perform the better than other classifiers. The results indicate that for the sampling applications, a RBM trained with QA is as good as a RBM trained with CD. Based on the classification accuracy results, we infer that both scheme 1 and scheme 2 significantly improved the classification accuracy compared to the case when the dataset was imbalanced. The learning of QA-based RBM can be improved with the availability of improved quantum annealers with a large number of qubits as well as by using an efficient procedure to determine the effective temperature of the QPU instead of treating it as a hyperparameter S.

Acknowledgment

We are grateful for the support from Integrated Data Science Initiative Grants (IDSI F.90000303), Purdue University. S.K. would like to acknowledge funding by the U.S. Department of Energyhttps://doi.org/10.13039/100000015 (Office of Basic Energy Sciences) under Award No. DE-SC0019215. T.S.H. would like to acknowledge funding by the U.S. Department of Energyhttps://doi.org/10.13039/100000015, Office of Basic Energy Science. This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725. We would like to thank D-Wave RFP award under project 'USRA NASA-AMES System Time'. This manuscript has been released as a pre-print at arXiv:2011.13996 [quant-ph] [44].

This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for

the United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan.

Author contributions The research was planned by S. Kais, Y. Koshka, M.A. Novotny, T.S. Humble, and M.A. Alam. Algorithm development and calculations were performed by Vivek Dixit and Raja Selvarajan. Tamer Aldwairi modified the original ISCX dataset into a 64-bit binary dataset. All authors contributed to the discussion of results and the writing of the manuscript.

REFERENCES

[1] M. Smith, Zhanna, and E.Lostri, "The hidden costs of cybercrime," Tech. Rep. Santa Clara: McAfee., 2020. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-hidden-costs-of-cybercrime.pdf

[2] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.

[3] T. Aldwairi, D. Perera, and M. A. Novotny, "An evaluation of the performance of restricted boltzmann machines as a model for anomaly network intrusion detection," *Comput. Netw.*, vol. 144, pp. 111–119, 2018.

[4] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *Proc. Nat. Aerosp. Electron. Conf.*, 2015, pp. 339–344.

[5] M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," in *Proc. Soft Comput. Ind. Appl.*, 2011, pp. 293–303.

[6] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *Int. J. Secur. Appl.*, vol. 9, no. 5, pp. 205–216, 2015.

[7] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl.*, 2016, pp. 195–200.

[8] A. Mizel, D. A. Lidar, and M. Mitchell, "Simple proof of equivalence between adiabatic quantum computation and the circuit model," *Phys. Rev. Lett.*, vol. 99, no. 7, 2007, Art. no. 0 70502.

[9] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution," 2000, *arXiv:quant-ph/0001106*.

[10] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, "Adiabatic quantum computation is equivalent to standard quantum computation," *SIAM J. Comput.*, vol. 37, no. 1, pp. 166–194, 2007.

[11] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse ising model," *Phys. Rev. E*, vol. 58, pp. 5355–5363, Nov. 1998.

[12] A. Finnila, M. Gomez, C. Sebenik, C. Stenson, and J. Doll, "Quantum annealing: A new method for minimizing multidimensional functions," *Chem. Phys. Lett.*, vol. 219, no. 5, pp. 343–348, 1994.

[13] T. Potok *et al.*, "Adiabatic quantum linear regression," 2020, *arXiv:2008.02355*.

[14] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen, "Support vector machines on the D-wave quantum annealer," *Comput. Phys. Commun.*, vol. 248, 2020, Art. no. 107006.

[15] V. Kumar, G. Bass, C. Tomlin, and J. Dulny, "Quantum annealing for combinatorial clustering," *Quantum Inf. Process.*, vol. 17, no. 2, p. 39, 2018.

[16] S. Das, A. J. Wildridge, S. B. Vaidya, and A. Jung, "Track clustering with a quantum annealer for primary vertex reconstruction at hadron colliders," 2019, *arXiv:1903.08879*.

[17] D. Arthur *et al.*, "Balanced k-means clustering on an adiabatic quantum computer," 2020, *arXiv:2008.04419*.

[18] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, and S. Kais, "Quantum annealing for prime factorization," *Nature Sci. Rep.*, 2018.

[19] R. Xia, T. Bian, and S. Kais, "Electronic structure calculations and the ising hamiltonian," *J. Phys. Chem. B*, vol. 122, no. 13, pp. 3384–3395, 2018.

[20] S. H. Adachi and M. P. Henderson, "Application of quantum annealing to training of deep neural networks," 2015, *arXiv:1510.06356*.

[21] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, "Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning," *Phys. Rev. A*, vol. 94, Aug. 2016, Art. no. 0 22308.

[22] Y. Koshka and M. A. Novotny, "Toward sampling from undirected probabilistic graphical models using a D-wave quantum annealer," *Quantum Inf. Process.*, vol. 19, no. 10, pp. 1–23, 2020.

[23] Y. Koshka and M. A. Novotny, "Comparison of D-wave quantum annealing and classical simulated annealing for local minima determination," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 2, pp. 515–525, Aug. 2020.

[24] V. Dixit, R. Selvarajan, M. A. Alam, T. S. Humble, and S. Kais, "Training and classification using a restricted boltzmann machine on the D-wave2000Q," 2020, *arXiv:2005.03247*.

[25] J. Caldeira, J. Job, S. H. Adachi, B. Nord, and G. N. Perdue, "Restricted boltzmann machines for galaxy morphology classification with a quantum annealer," 2019, *arXiv:1911.06259*.

[26] J. Sleeman, J. Dorband, and M. Halem, "A hybrid quantum enabled RBM advantage: Convolutional autoencoders for quantum image compression and generative learning," in *Quantum Inf. Sci., Sens., Comput. XII*, vol. 11391. International Society for Optics and Photonics, 2020, Art. no. 113910B.

[27] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.

[28] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[29] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, 2018.

[30] X. Ma and W. Shi, "Aesmote: Adversarial reinforcement learning with smote for anomaly detection," *IEEE Trans. Netw. Sci. Eng.*, 2020, doi: 10.1109/TNSE.2020.3004312.

[31] B. Yan, G. Han, M. Sun, and S. Ye, "A novel region adaptive smote algorithm for intrusion detection on imbalanced problem," in *Proc. 3rd IEEE Int. Conf. Comput. Commun.*, 2017, pp. 1281–1286.

[32] P. Su, Y. Liu, and X. Song, "Research on intrusion detection method based on improved smote and XGBoost," in *Proc. 8th Int. Conf. Commun. Netw. Secur.*, 2018, pp. 37–41.

[33] A. Tesfahun and D. L. Bhaskari, "Intrusion detection using random forests classifier with smote and feature reduction," in *Proc. Int. Conf. Cloud Ubiquitous Comput. Emerg. Technol.*, 2013, pp. 127–132.

[34] M. Ahsan, R. Gomes, and A. Denton, "SMOTE implementation on phishing data to enhance cybersecurity," in *Proc. IEEE Int. Conf. Electro/Inf. Technol.*, 2018, pp. 0531–0536.

[35] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, 2012.

[36] I. Sutskever and T. Tieleman, "On the convergence properties of contrastive divergence," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 789–795.

[37] V. Dumoulin, A. C. IanJ. Goodfellow, and Y. Bengio, "On the challenges of physical implementations of RBMs," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/viewPaper/8608

[38] T. D. Goodrich, B. D. Sullivan, and T. S. Humble, "Optimizing adiabatic quantum program compilation using a graph-theoretic framework," *Quantum Inf. Process.*, vol. 17, Apr. 2018. [Online]. Available: https://doi.org/10.1007/s11128-018-1863-4

[39] Y. Koshka, D. Perera, S. Hall, and M. A. Novotny, "Determination of the lowest-energy states for the model distribution of trained restricted Boltzmann machines using a 1000 Qubit D-wave 2X quantum computer," *Neural Comput.*, vol. 29, no. 7, pp. 1815–1837, 2017.

[40] Y. Koshka and M. A. Novotny, "Comparison of use of a 2000 Qubit D-wave quantum annealer and MCMC for sampling, image reconstruction, and classification," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 1, pp. 119–129, Feb. 2021.

[41] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[42] I. H. Witten and E. Frank, "Data mining: Practical machine learning tools and techniques with java implementations," *ACM SIGMOD Rec.*, vol. 31, no. 1, pp. 76–77, 2002.

[43] T. Aldwairi, D. Perera, and M. A. Novotny, "Measuring the impact of accurate feature selection on the performance of RBM in comparison to state of the art machine learning algorithms," *Electronics*, vol. 9, no. 7, 2020, Art. no. 1167.

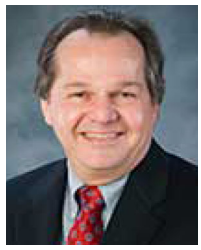
[44] V. Dixit, R. Selvarajan, M. A. Alam, T. S. Humble, and S. Kais, "Training a quantum annealing based restricted boltzmann machine on cybersecurity data," 2020, *arXiv:2011.13996*.

Vivek Dixit received the M.S. and Ph.D. degrees in physics and applied physics from Mississippi State University, Starkville, MS, USA, in 2011 and 2015, respectively. He is currently a Postdoctoral Research Associate with the Department of Chemistry, Purdue University, West Lafayette, IN, USA. His research interests include quantum machine learning, quantum computing, electronic structure calculations of solids and molecular systems, and spectroscopy.

Raja Selvarajan is currently working toward the Graduation degree with the Department of Physics and Astronomy, Purdue University, West Lafayette, IN, USA. His research interests include quantum machine learning and quantum computing.

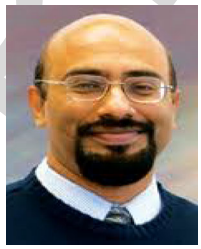
Tamer Aldwairi received the M.S. degree in computer science and the Ph.D. degree in computational engineering from Mississippi State University, Starkville, MS, USA, in 2014. From 2015 to 2017, he was a Postdoc Research Associate with the Distributed Analytics and Security Institute, High-Performance Computing Collaboratory. From 2017 to 2019, he was a Visiting Assistant Professor with Ursinus College, Collegeville, PA, USA. He is currently an Assistant Professor of instruction with Temple University, Philadelphia, PA, USA. His research interests include machine learning, big data analytics, cybersecurity, quantum computing, and high-performance computing.

Yaroslav Koshka received the B.S. and M.S. degrees in electronics from Kiev Polytechnic University, Kiev, Ukraine, in 1993, and the Ph.D. degree in electrical engineering from the University of South Florida, Tampa, FL, USA, in 1998. He is currently a Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS, USA, and the Director of Emerging Materials Research Laboratory. His current main research focuses on quantum computations, their application to machine learning and properties of electronic materials and other research interests include semiconductor materials and device characterization, defect engineering, synthesis of wide-bandgap semiconductor materials and nanostructures, physics of semiconductor devices, and nanoelectronics.



Mark A. Novotny received the B.S. degree in physics from North Dakota State University, Fargo, ND, USA, in 1973 and the Ph.D. degree in physics from Leland Stanford Junior University, Stanford, CA, USA, in 1979. Since 2001, he has been a Professor and the Head of the Department of Physics and Astronomy, Mississippi State University, Starkville, MS, USA. He has authored or coauthored more than 200 papers in refereed journals. His research interests include classical and quantum computational physics, with applications to classical and quantum properties of materials and statistical mechanics. He is a Giles Distinguished Professor with Mississippi State University, a Fellow of the American Physical Society, and a Fellow of AAAS.

Travis S. Humble received the Doctorate degree in theoretical chemistry from the University of Oregon, Eugene, OR, USA, in 2005. He is currently the Deputy Director of the Department of Energy's Quantum Science Center, a Distinguished Scientist with Oak Ridge National Laboratory, Oak Ridge, TN, USA, and the Director of lab's Quantum Computing Institute. He is leading the development of new quantum technologies and infrastructure to impact the DOE mission of scientific discovery through quantum computing. He is the Editor-in-Chief of the *ACM Transactions on Quantum Computing,* an Associate Editor for the *Quantum Information Processing*, and the Co-Chair of the IEEE Quantum Initiative. He also holds a joint faculty appointment with the University of Tennessee Bredesen Center for Interdisciplinary Research and Graduate Education, where he works with students in developing energy-efficient computing solutions.



Muhammad A. Alam (Fellow, IEEE) is the Jai N. Gupta Professor of electrical engineering with Purdue University, West Lafayette, IN, USA, where his research focuses on physics, simulation, characterization and technology of classical and emerging electronic devices, including reliability of scaled MOSFET, theoretical foundation of nano-bio sensors, and atom-to-farm performance and modeling of solar cells. He was the recipient of the 2006 IEEE Kiyo Tomiyasu Medal, 2015 SRC Technical Excellence Award, and 2018 IEEE EDS Education Award.

Sabre Kais received the B.Sc., M.Sc., and Ph.D. degrees from The Hebrew University of Jerusalem, Jerusalem, Israel, in 1983, 1984, and 1989, respectively. From 1989 to 1994, he was a Research Associate with the Department of Chemistry, Harvard University, Cambridge, MA, USA. In 1994, he joined Purdue University, West Lafayette, IN, USA, as an Assistant Professor of theoretical chemistry. He is currently a Full Professor of chemical physics, a Professor of computer science (courtesy), a Professor of physics with Purdue University. He has authored or coauthored more than 240 papers in peer-reviewed journals. His research interests mainly include finite size scaling, dimensional scaling, quantum information, and quantum computing, From 2010 to 2013, he was the Director of NSF funded center of innovation on Quantum Information for Quantum Chemistry. He is a Fellow of the American Physical Society, a Fellow of the American Association for the Advancement of Science, Guggenheim Fellow, Purdue University Faculty Scholar, and National Science Foundation Career Award Fellow. He was the recipient of the 2012 Sigma Xi Research Award, and 2019 Herbert Newby McCoy Award, Purdue University.