Pan-cancer Feature Selection and Classification Reveals Important Long Non-coding RNAs

Abdullah Al Mamun
School Computing and Information Sciences
Florida International University
Miami, FL, USA
mmamu009@fiu.edu

Wenrui Duan
Department of Human & Molecular Genetics
Herbert Wertheim College of Medicine
Florida International University
Miami, FL, USA
wduan@fiu.edu

Ananda Mohan Mondal*
School of Computing and Information Sciences
Florida International University
Miami, FL, USA
amondal@fiu.edu

Abstract- Long noncoding RNA plays important role in changing the expression profiles of various target genes that leads to cancer development. So, identifying key lncRNAs related to the origin of different types of cancers might help in developing cancer therapy. To discover the critical lncRNAs that can identify the origin of different cancers, we proposed to use the state-of-the-art deep learning algorithm Concreate Autoencoder (CAE). The motivation behind using the CAE was that it takes advantage of both AE (which can achieve the highest classification accuracy) and concrete relaxation-based feature selection (which is capable of selecting actual features instead of latent features). To compare the performance of CAE, three frequently used embedded feature selection techniques including Least Absolute Shrinkage and Selection Operator (LASSO), Random Forest (RF), and Support Vector Machine with Recursive Feature Elimination (SVM-RFE) were used. To obtain a stable set of lncRNAs capable of identifying the origin of 33 different cancers, a lncRNA that was isolated by at least two of the four techniques (CAE, LASSO, RF, and SVM-RFE) was added to the final list of key lncRNAs.

The genome-wide lncRNA expression profiles of 33 different types of cancers, a total of 9566 samples, available in The Cancer Genome Atlas (TCGA) were analyzed to discover the key lncRNAs. Our results showed that CAE performs better in feature selection, specially, in selecting small number of features, compared to LASSO, RF, and SVM-RFE. With the increasing number of selected features ranging from 10 to 500 lncRNAs, the accuracy of different feature selection approaches increases as - CAE: 70% to 96%; LASSO: 55% to 94%; RF: 38% to 95%; SVM-RFE: 50% to 94%. This study discovered a set of 69 lncRNAs that can identify the origin of 33 different cancers with an accuracy of 93%. Note that the accuracy could be higher using AE, which uses latent features for classification thus failing to correlate the origin of cancers with the actual features (lncRNAs).

The proposed computational framework can be used as a diagnostic tool by the physicians to discover the origin of cancers using the expression profiles of lncRNAs. The discovered lncRNAs can be studied further by biologists or drug designer to identify possible targets for cancer therapy.

Keywords – Autoencoder, Concrete Autoencoder, Deep learning, Feature Selection, lncRNA.

I. INTRODUCTION

Recent studies indicate that several cancer risk loci are transcribed into long non-coding RNAs (lncRNAs) and these transcripts play key roles in tumorigenesis [1], [2]. The lncRNAs also have key functions in transcriptional, post-transcriptional, and epigenetic gene regulation [3]. Schmitt *et al.* discussed the impact of lncRNA in cancer pathways [4]. They described the involvement of lncRNAs in six hallmarks of cancer such as proliferation, growth suppression, motility, immortality, angiogenesis, and viability [5].

Hoadley *et al.* showed that cell of origin patterns dominate the molecular classification of tumors available in The Cancer Genome Atlas (TCGA) [6]. For their analysis, they used copy number, mutation, DNA methylation, RPPA protein, mRNA, and miRNA expression. But they did not consider another important molecular signature of cancer, which is lncRNA expression. This work motivated us to investigate the importance of lncRNAs in identifying cancer origins.

Though RNAseq data from TCGA contains a reasonable number of samples, even it poses challenges for classification tasks due to a large number of features (lncRNAs) with respect to the number of samples. Many computational methods fail to identify a small number of relevant features, rather increase learning costs and deteriorate performance [7]. It may be argued that the larger the feature set, the better the classification. However, in a general setting, not all of these features will be necessary for optimal classification. Only a selected number of significant or relevant features can lead to optimal classification. A large part of the remaining features are not significant and could be either noise, irrelevant to the study, or even redundant [8]. The use of such insignificant features can lead to unwanted computational complexities and deteriorate the performance of the model. This is more pronounced when working with high-dimensional data. Thus, it is essential to identify the set of significant features that can provide us with the optimal classification and clustering. To accomplish this objective, we need a robust method that can eliminate the redundant features and noise that do not carry any information about the labels of data, thus providing us with only relevant features [9].

Any dataset with N-number of features has 2^N-possible subset of features [8]. In the presence of such a large number of possible combinations, finding the best subset of N features is computationally challenging and expensive [10]. An optimally selected set of features not only optimizes the performance of

classification models but also helps in alleviating the effect of overfitting and high-dimensionality. Along with these benefits, selecting the appropriate features helps in the easier interpretation of the model as well as its predictions. On the other hand, the use of gratuitous features can significantly impact the training speeds and the accuracy of the learning models.

Filter, wrapper, embedded methods are the three general classes or types of feature selection techniques. The filtering method works by ranking the features using a statistical score that is assigned to each of them depending on their relevance to the class type. In both univariate and multivariate filter methods, the interactions among features are disregarded in the selection process. Studies like the ones in Pearson correlation coefficient(PCC), t-statistics(TS) [11], F-Test [12], and ANOVA [13] are examples where the filtering method is used. It is observed that these methods are effective for selecting features in high-dimensional data because of the reduced computation expenses. However, they fail to provide good accuracy as discussed in [14].

As an enhancement, the researcher developed the wrapper-based feature selection method with a learning algorithm and a classifier to find a suitable subset of features. Initially, a random solution is generated, following which, an objective function is maximized using black-box type optimization methods [15] like simulated annealing [16], particle swarm optimization [17], genetic algorithm [18], and ant colony optimization [19]. The iterative evaluation of every candidate subset of the features by a wrapper method leads to the identification of a strong relationship between features, however with an increase in the computational expense.

Embedded feature selection methods on the other hand reduces computational costs because these are used as a part of the learning phase. Well-known embedded methods, which are considered as the state-of-the-art, are least absolute shrinkage and selection operator (LASSO) [20], recursive feature elimination with support vector machine estimator (SVM-RFE) [21]–[23], random forest [24], [25], Adaboost [26] , KNN [27] , and autoencoder [28].

In general, the use of feature selection is worthwhile when the whole set of features is difficult to collect or expensive to generate [34].. For example, in TCGA, the lncRNA expression profile dataset contains more than 12 thousand features (lncRNAs) for each of 33 different cancers and it is expensive to generate this data. Consequently, it is important to answer the question: *Is there a set of salient features (lncRNAs) capable of identifying the origin of 33 cancers?*

The distribution of number of samples for 33 cancers in TCGA is highly imbalanced, ranging from 36 for CHOL cancer to 1089 for BRCA. Any supervised feature selection approach will be biased to heavy groups. To solve this problem, we need a robust unsupervised feature selection approach capable of finding appropriate features related to 33 different cancers.

Feature selection works differently compared to the standard dimension reduction techniques such as principal component analysis (PCA) [29], and autoencoders [30]. The standard dimension reduction methods can preserve maximum variance with a highly reduced number of latent features. This means that PCA and standard autoencoder do not provide the original features in the reduced dimension or these work as a black-box. For real application of diagnosing the origin of cancer, a tool should be able to tell what actual or measurable features are relevant. Recently, few deep learning-based feature selection methods showed little improvement in selecting original features in both settings supervised and unsupervised [31]–[33].

In this paper, we proposed to use concrete autoencoder (CAE) [34], a deep learning-based unsupervised feature selection algorithm, to discover the relevant lncRNAs that are capable of identifying the origin of different cancers. The CAE takes advantage of both (a) AE, which can achieve the highest classification accuracy and (b) concrete relaxation-based feature selection [35], [36], which is capable of selecting actual features instead of latent features. Proposed model filtered the key lncRNAs from 12,309 lncRNAs, that are related to 33 different cancers. The key lncRNAs discovered using the proposed CAE method produced higher classification accuracy and better diagnosis of cancer origin compared to the state-of-the-art embedded feature selection approaches – LASSO, RF, and SVM-RFE - while using small number of lncRNAs.

II. MATERIALS AND METHODS

The overall process flow diagram is illustrated in Figure 1. The following subsections describe the different aspects of process flow diagram: (a) Data Preparation, (b) Feature Selection, (c) Reconstruction and Classification, and (d) Evaluation and Validation.

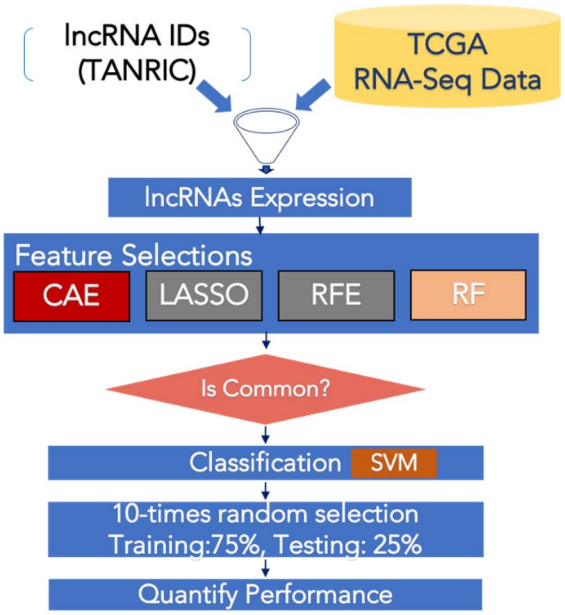

Fig. 1: Process flow diagram. Data Preparation, Feature Selection, Classification and Validation.

A. *Data Preparation*

To characterize the cancer-associated lncRNA, expression profiles and clinical data for 33 different cancers were downloaded from UCSC Xena database [37] . This dataset contains expression profiles of about 60 thousand RNAs including coding genes (mRNAs) and non-coding genes (lncRNAs and miRNAs). In this study, only the expression

profiles of lncRNA (n = 12,309) were considered for analysis and model evaluation. It should be noted that this study was based on cancer patients only. So, normal samples available in the same cancer were removed. The final dataset contains 9,566 cancer patients. The cancer-specific distributions based on 75/25 (training/testing) split are shown in Fig. 2. Each lncRNA expression was processed using a min-max normalization method to achieve good training performance.

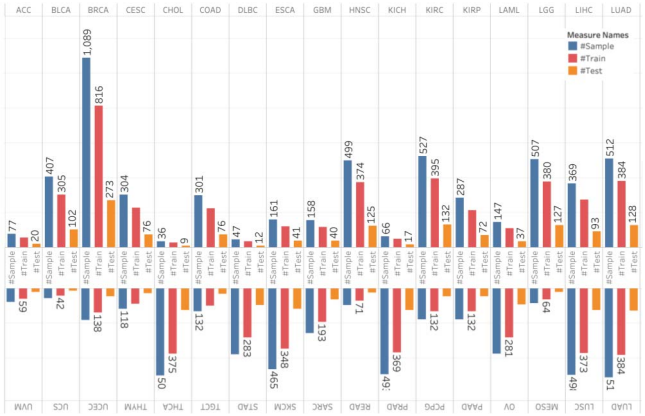

Fig. 2: Sample distribution for 33 cancers along with 75-25 split for training and testing.

B. *Features Selection*

For selecting important features (lncRNAs), a state-of-the-art deep learning-based unsupervised algorithm, Concreate Autoencoder (CAE), was used. To compare the results of CAE, 3 frequently used embedded feature selection models, including LASSO, Random Forest (RF), and Support Vector Machine with Recursive Feature Elimination (SVM-RFE), were used. Following subsections briefly describe the implementation of feature selection algorithms.

1) *Concrete Autoencoder (CAE)*

Concrete autoencoder (CAE) proposed by Abid *et al.* [34] is a variation of the original autoencoder (AE) [30], which is used for dimension reduction. The motivation behind selecting CAE in the present study is that it takes advantage of both AE (which can achieve the highest classification accuracy) and concrete relaxation-based feature selection (which is capable of selecting actual features instead of latent features). An AE is a neural network that consists of two parts: (a) an encoder that selects latent features and (b) a decoder that uses selected latent features to reconstruct an output that matches the input with minimum error. In CAE, instead of using a sequence of fully connected layers in the encoder, a concrete relaxation-based feature selection layer is used where the user can define the number of nodes (features to be selected), k as shown in Fig. 3. This layer selects a probabilistic linear arrangement of input features while training, which converges to a discrete set of k features by the end of training phase, which are subsequently used in the testing phase.

Let's $p(x)$ is a probability distribution over a d-dimensional vector. The objective is to identify a subset of features, S≡{1…k} of size $|S|=k$. Also, learning a reconstruction function $f_r(.)$: $\mathbb{R}^k \xrightarrow{\Delta} \mathbb{R}^d$, such that the loss between original sample $\boldsymbol{x}$ and reconstructed sample $f_r(x_S)$ is minimized as stated in Eq. 1,

$$argmin_{S,r}\, E_{p(x)}[\|f_r\,(x_S\,) - x\|_2] \ldots\ldots\ldots\ldots\ldots (1)$$

where $x_s \in \mathbb{R}^k$ consists of only selected features x_i s.t. $i \in S$. Note that samples are represented in a *2D* matrix, $\boldsymbol{X}\in\mathbb{R}^{n\times d}$, and aim is to pick k columns of $\boldsymbol{X}$ such that sub-matrix $X_s \in \mathbb{R}^{n\times k}$.

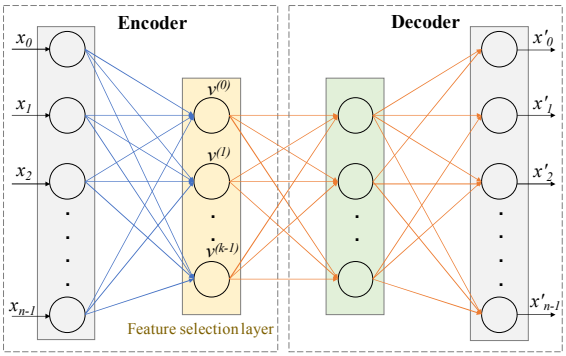

Fig. 3: Architecture of Concrete Autoencoder. CAE architecture consists of an encoder and a decoder. The layer after input layer of encoder is called concrete feature selection layer shown in yellow. This layer has k number of node where each node is for each feature to be selected. During the training stage, the i^{th} node $v^{(i)}$ takes the value X^T *f(i)*, where *f(i)* is the corresponding weight vector of node *i*. During testing stage, these weights are fixed and the element with the highest value is selected by the corresponding i^{th} hidden node. The architecture of the decoder remains the same during training and testing.

Then, selected feature set x_s can be used to reconstruct the original matrix $\boldsymbol{X}$ and classify the cancer types.

In feature selection layer of CAE, (Fig.3), the original features are selected based on the temperature of this layer which is tuned using an annealing schedule. More specifically, the concrete selector layer identifies k important features as the temperature decreases to zero. For reconstructing the input, a simple decoder similar to the ones associated with a standard AE is used. The temperature τ of the random variable in the selector layer has a significant impact in forming the output of each node. Initially, when τ is high, search space is large, since it considers a linear combination of all features as shown in Fig. 4(a). In contrast, the selector layer will not be able to search all possible combinations of features at low τ and thus, the model converges to a poor local minimum. This means that as temperature goes down, small number of features are necessary for stable convergence. Annealing or gradual decrease in temperature avoids the model convergence to a poor local minimum. The effect of annealing in feature selection is shown in Fig. 4(a). For example, at starting temperature, τ_s, the number of input features

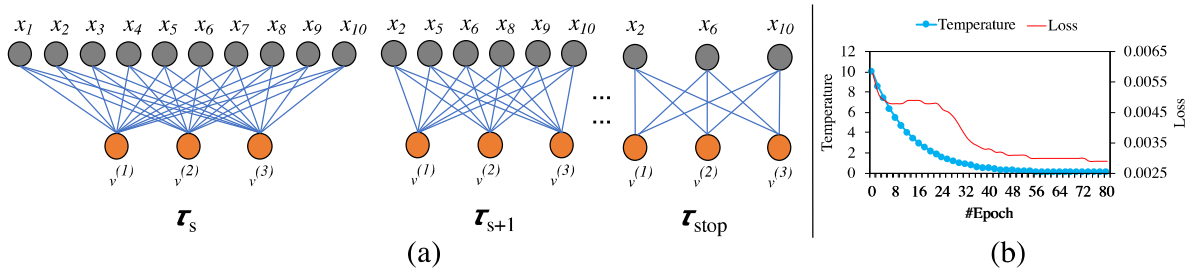

Fig. 4: Effect of annealing in reducing search space. (a) An example: at starting temperature τ_s, the number of input features is 10 and the number of features to be selected is k = 3; at the next epoch when the temperature is τ_{s+1}, the number of possible features reduces to 6; after some epochs, when the temperature reaches to its lower bound τ_{stop}, the number of features further reduces to 3, which is equal to k. (b) Effect of temperature change in reducing the loss while training the concrete autoencoder on lncRNA expression data with k = 100 features to be selected from original feature space of 12,309 lncRNAs.

is 10 and the number of features to be selected, k is 3. At the next epoch when the temperature is τ_{s+1}, the number of possible features reduces to 6. After some epochs, when the temperature reaches its lower bound τ_{stop}, the number of features further reduces to 3, which is equal to k, user-specified number of features to be selected. Instead of using a fixed temperature, a simple annealing scheduling scheme is used for every concrete variable. It starts with a user-defined high temperature (τ_s) and steadily lowers the temperature, until it touches the end bound (τ_e), by every epoch as follows:

$$\tau_{(e)} = \tau_s\,(\tau_N/\tau_s)^{e/n} \quad \text{.....(2)}$$

where τ_e is the temperature at epoch e, N refers to total number of epochs. Adam optimizer with a learning rate of 0.001 is used for all the experiments for CAE. Figure 4(b) shows an example of the effect of temperature in reducing the loss while training the CAE to select a reduced set of 100 features from the original feature space of 12,309 lncRNAs. The starting temperature of CAE was set to 10 and it ends at 0.01. To control the performance, the model was trained for the same number of epoch (n = 100).

2) Implementation of LASSO

To select the important features, LASSO applies a regularization (shrinking) process where it penalizes the coefficients of the regression variables and shrinks these to zero. The variables that still have a non-zero coefficient are selected as the top features. The tuning parameter λ controls the strength of the penalty. The larger is the parameter λ, the larger number of coefficients are shrunk to zero and smaller number of features are selected. In this experiment, the optimized λ was set in a range of 0.005 to 0.01 to select a different number of features ranging from 10 to 500.

3) Implementation of RF

Random Forest works based on tree structure that employs ensemble. RF consists of a number of decision trees. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two branches so that similar response values end up in the same set. The optimal condition is chosen based on impurity. For classification, it is either *Gini* impurity or information gain/entropy. Thus, when the tree is fully developed, it can compute how much each feature decreases the weighted impurity on the tree. For forest, the impurity decrease from each feature can be measured as a feature rank. The feature importance is calculated as the sum over the number of splits (across all trees) that include the feature, proportionally to the number of samples it splits [39]. RF needs three parameters to be tuned: (i) *n_estimator:* number of estimators, also known as number of trees in the forest, (ii) *min_sample_split:* minimum number of nodes required to split, and (iii) *criterion:* impurity to measure the quality of a split. In *GridSearch,* the ranges of values assigned to tune *n_estimator* and *min_sample_split* were from 2 to 300 and 1 to 150, respectively. Two options, *Gini* and *entropy*, were used to optimize the impurity parameter *criterion*. The optimum values or options for *n_estimator*, *min_sample_split, and criterion* found by *the GridSearch* method are 100, 120, and *Gini*, respectively.

4) Implementation of SVM-RFE

Recursive feature elimination is a recursive method in which less important features are eliminated in every iteration. In RFE technique, SVM was used as the estimator in the present study. A linear kernel with a regularization parameter C = 0.05 was used. C controls the tradeoff between the error and norm of the learning weights. The *GridSearch* algorithm was used to estimate the best set of parameters for SVM. In every iteration of RFE, the number of dropped features was set to 100.

LASSO, RF, and SVM-RFE were implemented using the scikit-learn framework [40] whereas CAE was implemented using TensorFlow [41] based deep learning framework, Keras [42]. Experiments are parallelized on NVIDIA Quadro K620 GPU with 384 cores and 2GB memory devices. To avoid overfitting, the dataset was split into the train and test set according to 75/25 ratio, as shown in Fig. 2. The training set was used to estimate the learning parameters and the test set was used for performance evaluation.

C. Reconstruction and Classification

The feature selection capability of CAE is compared with standard autoencoder (AE), LASSO, RF, and SVM-RFE in two different ways: (a) reconstruction of all input features using the selected features and (b) classification performance in classifying 33 different cancer types using the selected features. A subset of features by varying k from 10 to 500 were extracted using CAE. For the comparison to be fair and along the same grounds with CAE, the same number of lncRNAs were selected using all other models. The SVM was used for classifying 33 cancer types using the selected features. To reconstruct all the input features from the selected features, we trained a linear regressor with no regularization.

D. Evaluation and Validation

Five different evaluation metrics have been used to record the classification and reconstruction performance such as

accuracy, precision, recall, f1 score, and mean squared error (MSE). Accuracy is the number of correct predictions made by the model over all kinds of predictions made. Precision is the number of correct positive results divided by the number of positive results predicted by the model. It indicates the predicted positive portion of the samples. Recall is the number of correct positive results divided by the number of all relevant samples. F1 score is the harmonic mean of precision and recall. Reconstruction performance measure, MSE, was calculated using linear regression on the test set.

All classification performance metrices were measured by comparing the predicted labels with the true labels of independent test samples. The optimal set of features was selected based on two criteria: (a) the number of features should be as few as possible, and (b) classification accuracy using the selected features should be > 90%. The final list of key lncRNAs is selected from the union of features derived from the binary intersection of four approaches,

$$(CAE \cap LASSO) \cup (CAE \cap RF) \cup (CAE \cap SVMRFE) \cup (LASSO \cap RF) \cup (LASSO \cap SVMRFE) \cup (RF \cap SVMRFE) \quad \ldots\ldots(3)$$

Then each lncRNA discovered in this study was cross-checked with existing literature whether it is already a known biomarker or not. The capability of selected lncRNAs in pan-cancer classification was visually validated using unsupervised visualization technique t-SNE [43]. To validate the prognostic performance of discovered lncRNAs, survival analysis of cancer patients using Kaplan-Meier [44] method was performed [45].

III. RESULTS

A series of experiments were conducted to compare the performance of CAE with other state-of-the-art feature selection methods such as standard autoencoder, LASSO, RF, and SVM-RFE. Each of these methods was used to select features in the range of 10 to 500. These features were then used to train a linear classifier SVM to classify 33 cancer types using expression profiles of lncRNAs.

A. Classification Performance Using Selected Sets of Features

Fig. 5 shows classification performance using different sets of selected features. The initial stages of the experiments were performed with a smaller subset of the selected features as we wanted to understand the performance of the models being compared. The optimal classification performance with CAE (accuracy > 90% with the smallest number of features) was observed with about 100 features. Beyond this point, the increase in performance was not significant.

It is clear from Fig. 5 that, for all sets of selected features, CAE performed better than LASSO, RF, and SVM-RFE in terms of four evaluation matrices, including accuracy, precision, recall, and f1 score. Of course, it could not beat the standard AE, as expected. It is noticeable that even with a smaller number of features (say 10), the accuracy of CAE was close to 70%, whereas LASSO (55% accuracy), RF (38% accuracy), and SVM-RFE (50% accuracy) showed poor results for the same number of features. The trend remains the same with the increase of number of features.

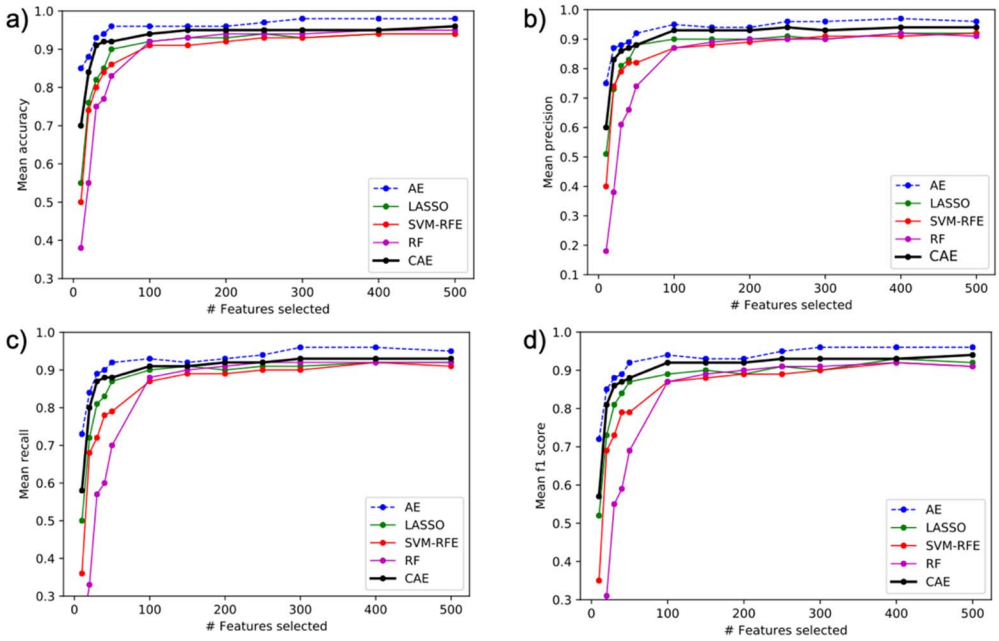

Fig. 1: Classification performances of proposed method using selected features. Comparison of CAE with other feature selection methods. Throughout the all values of *k* tested on both (a) Accuracy, (b) Precision, c) Recall, and d) f1 score; CAE have highest classification performance after AE.

B. *Reconstruction Performance of Feature Selection Algorithms*

Figure 6 shows the comparison of reconstruction performance among five feature selection algorithms. Note that AE select latent features, whereas, other four algorithms select actual features. The CAE starts with an MSE of 60 and quickly reduces to a value of less than 10 within the use of the top 100 features as shown in Fig. 6. It is also clear from this figure that CAE has lower reconstruction error compared to LASSO, RF, and SVM-RFE for any set of selected features. Again, CAE cannot

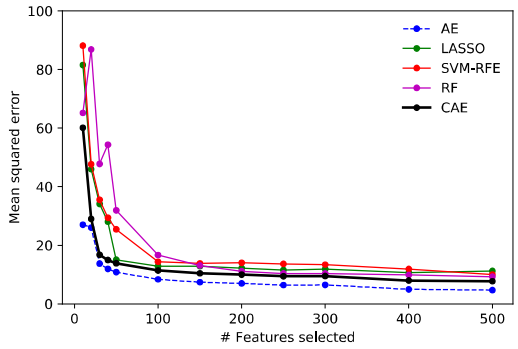

Fig. 2: Reconstruction mean squared error for different feature sizes selected by different models

C. *Combined Features*

Based on the performance of CAE, a set of 100 lncRNAs (features) produced an optimal classification. So, to produce a stable set of features for this problem, each of the four feature selection algorithms were run to extract 100 features. The final list of 69 key lncRNAs was the result of the union of features derived from the binary intersection of four approaches as mentioned in eq. 3. Figure 7 shows the Venn diagram of the common features extracted by four algorithms. It is clear from the Venn diagram that 67 (100 – 23) out of 69 lncRNAs came from CAE, which dictates the superiority of CAE.

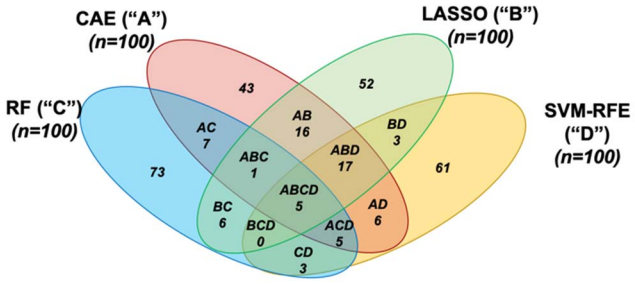

Fig. 7: Common features selected by different methods

Table I shows the comparison of classification (Accuracy, Precision, Recall, and f1) and reconstruction (MSE) performance among the four approaches. Selected 100 features from each method was passed to a linear regressor for reconstructing the input features. Performance using the combined feature set of 69 lncRNAs is also shown in the table. It is clear from this table and Fig. 6 that CAE is more resilient to reconstruction error, whereas, the error is more pronounced in the other competing methods. It is also clear from this table and Fig. 6 that CAE outperforms other state-of-the-art feature selection approaches. But it is noticeable that combined 69 features has better performance compared to the results produced by 100 features selected by three shallow feature selection approaches (LASSO, RF, and SVM-RFE). Not only that the combined 69 lncRNAs performs at the same level of CAE with 100 lncRNAs (93% accuracy). Of 69 combined features, 67 are coming from the 100 lncRNAs selected by CAE (Fig. 7). This means that a considerable number of lncRNAs (~30 lncRNAs) are not contributing in classification, which demand further investigation.

TABLE I: Classification and reconstruction performances using combined 69 lncRNAs and selected 100 lncRNAs using different models.

Model	Accuracy	Precision	Recall	F1	MSE
Combined	0.93±0.02	0.91±0.01	0.91±0.02	0.9±0.03	13.46±0.10
LASSO	0.92±0.01	0.87±0.02	0.88±0.02	0.87±0.01	13.84±0.08
SVM-RFE	0.85±0.03	0.85±0.02	0.82±0.03	0.83±0.02	25.98±0.08
RF	0.89±0.02	0.86±0.03	0.81±0.03	0.81±0.03	22.91±0.12
CAE	0.93±0.01	0.89±0.01	0.9±0.02	0.9±0.02	12.23±0.09

D. *Visual Validation of Selected Features*

Fig. 8 shows the unsupervised clustering capability of expression profiles of discovered 69 lncRNAs using the t-SNE plot [43]. It is clear from the t-SNE plot that the discovered lncRNAs are capable of discovering the heterogeneity among 33 cancers. So, the newly identified lncRNAs can be considered as essential features for diagnosis, prognosis, and therapeutic target for different cancers. Then each lncRNA was cross-checked with the existing literature whether it is already a known biomarker. Of the 69 lncRNAs, 38 were found in existing literature as known biomarkers for different cancers as shown in Table 2. The remaining 31 lncRNAs were novel discovery based on the ***lncRNA disease*** **database v2.0.**

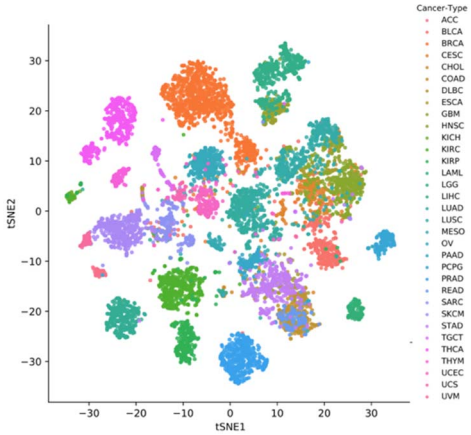

Fig. 8: t-SNE using top 69 lncRNAs where each dot represents a cancer sample and each color represents a cancer type.

IV. DISCUSSION

It is clear from the literature that lncRNAs play a key role in cancer development. More research is needed to identify cancer-specific lncRNAs. Existing methods used co-expression networks such as lncRNA-mRNA or lncRNA-miRNA-mRNA. As per our knowledge, no study used lncRNA expression only to

classify cancer types except our previous work [46] where feature extraction was not considered.

In this study, we identified 69 key lncRNAs that can identify the origins of 33 different cancers. When compared against the existing literature, 38 (55%) lncRNAs have been reported as important prognostic biomarkers for various cancers, Table II. Since the proposed method can identify already known lncRNA biomarkers, it can be concluded that the newly discovered 31

TABLE II: 69 key lncRNAs identified in this study

Known lncRNAs (n=38)
AC005083.1, AC008268.1, AC093850.2, AC133528.2, AFAP1-AS1, CASC9, CRNDE, DNM3OS, EMX2OS, FAM83H-AS1, FENDRR, GATA2-AS1, GATA6-AS1, H19, HAGLR, HAND2-AS1, HCG11, HNF1A-AS1, LHFPL3-AS1, LINC00261, LINC00511, LINC01116, LINC01133, LINC01139, LINC01158, MALAT1, MEG3, MNX1-AS1, NR2F1-AS1, PIK3CD-AS2, PTCSC2, SATB2-AS1, SFTA1P, TRPM2-AS, UCA1, VPS9D1-AS1, XIST, ZNF667-AS1
Novel lncRNAs (n=31) Based on *lncRNA disease* v2.0 (http://www.rnanut.net/lncrnadisease/) dated: July 2020
AC005082.12, AC079630.4, AP001626.1, CECR7, CTA-384D8.31, CTD-2377D24.4, CTD-3032H12.2, GATA3-AS1, HOXA10-AS, HOXA11-AS, HOXD-AS2, LINC00958, LINC01082, LINC01272, MIR205HG, NKX2-1-AS1, RP1-288H2.2, RP1-60O19.1, RP11-1017G21.5, RP11-1055B8.3, RP11-264B14.2, RP11-3P17.5, RP11-465B22.8, RP11-47A8.5, RP11-807H17.1, RP3-416H24.1, SLCO4A1-AS1, TBX5-AS1, U47924.27, U91324.1, ZFPM2-AS1

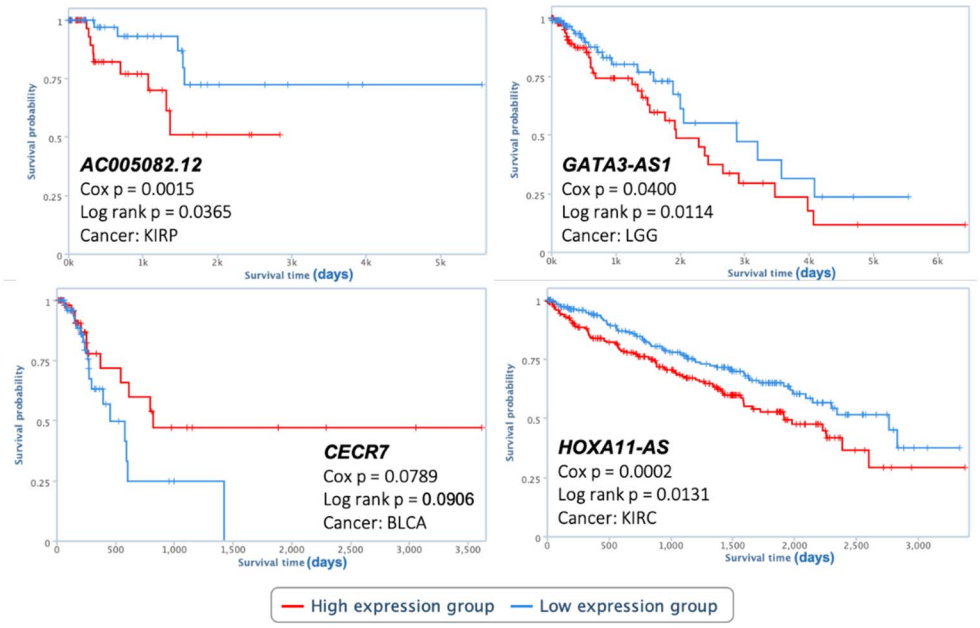

Fig. 9: Kaplan-Meier survival analysis curve of high-risk and low-risk patients evaluated on novel lncRNA (*AC005082.12, CECR7, GATA3-AS1, and HOXA11-AS*).

lncRNAs have the potential to be considered as novel biomarkers for cancers. Survival analysis suggests that some of 31 lncRNAs are novel biomarkers as shown in Fig. 9. Many studies have been conducted using mRNA expression for predicting cancer types as well as developing screening tools. No such tools are available that used expression profiles of lncRNAs. Hence, the identified 69 lncRNAs can be used not only as a screening tool for cancer diagnosis but also as therapeutic targets for different cancers, for which further studies are required.

V. CONCLUSION

In this paper, a computational framework was developed using concrete autoencoder, a deep learning-based unsupervised feature selection algorithm, to identify the key features. The proposed method was evaluated in identifying the origin of 33 different cancers using the expression profiles of selected features (69 lncRNAs) from the original feature space of 12 thousand lncRNAs. Existing literature and validation support that the selected lncRNAs could be potential biomarkers for diagnosis and prognosis of 33 different cancers. This paper accounts for both feature selection and identifying the origin of different cancers. However, other avenues can also be explored using the proposed method. For example, the proposed method can be used to identify important genes while classifying patients of a single cancer into molecular subtypes. It can also be used to integrate multi-omics data such as both coding and non-coding RNA expression as well as DNA methylation.

ACKNOWLEDGEMENT

This research was partially funded by NSF CAREER award #1651917 (transferred to #1901628) and NSF RAPID award #2037374.

REFERENCES

[1] S. W. Cheetham, F. Gruhl, J. S. Mattick, and M. E. Dinger, “Long noncoding RNAs and the genetics of cancer,” *Br. J. Cancer*, vol. 108, no. 12, pp. 2419–2425, Jun. 2013, doi: 10.1038/bjc.2013.233.

[2] Y. Fang and M. J. Fullwood, “Roles, Functions, and Mechanisms of Long Non-coding RNAs in Cancer,” *Genomics. Proteomics*

Bioinformatics, vol. 14, no. 1, pp. 42–54, Feb. 2016, doi: 10.1016/j.gpb.2015.09.006.

[3] H. Tao, J.-J. Yang, X. Zhou, Z.-Y. Deng, K.-H. Shi, and J. Li, "Emerging role of long noncoding RNAs in lung cancer: Current status and future prospects," *Respir. Med.*, vol. 110, pp. 12–19, Jan. 2016, doi: 10.1016/j.rmed.2015.10.006.

[4] A. M. Schmitt and H. Y. Chang, "Long Noncoding RNAs in Cancer Pathways," *Cancer Cell*, vol. 29, no. 4, pp. 452–463, Apr. 2016, doi: 10.1016/j.ccell.2016.03.010.

[5] D. Hanahan and R. A. Weinberg, "Hallmarks of Cancer: The Next Generation," *Cell*, vol. 144, no. 5, pp. 646–674, 2011, doi: https://doi.org/10.1016/j.cell.2011.02.013.

[6] K. A. Hoadley *et al.*, "Cell-of-origin patterns dominate the molecular classification of 10,000 tumors from 33 types of cancer," *Cell*, vol. 173, no. 2, pp. 291–304, 2018.

[7] L. Sun, X. Kong, J. Xu, R. Zhai, S. Zhang, and others, "A Hybrid Gene Selection Method Based on ReliefF and Ant Colony Optimization Algorithm for Tumor Classification," *Sci. Rep.*, vol. 9, no. 1, p. 8978, 2019.

[8] J. Pirgazi, M. Alimoradi, T. E. Abharian, and M. H. Olyaee, "An Efficient hybrid filter-wrapper metaheuristic-based gene selection method for high dimensional datasets," *Sci. Rep.*, vol. 9, no. 1, pp. 1–15, 2019.

[9] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*, vol. 454. Springer Science & Business Media, 2012.

[10] S. Liang, A. Ma, S. Yang, Y. Wang, and Q. Ma, "A review of matched-pairs feature selection methods for gene expression data analysis," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 88–97, 2018.

[11] T. Speed, *Statistical analysis of gene expression microarray data*. Chapman and Hall/CRC, 2003.

[12] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *J. Bioinform. Comput. Biol.*, vol. 3, no. 02, pp. 185–205, 2005.

[13] H. Ding and D. Li, "Identification of mitochondrial proteins of malaria parasite using analysis of variance," *Amino Acids*, vol. 47, no. 2, pp. 329–333, 2015.

[14] Y. Sun, C. Lu, and X. Li, "The cross-entropy based multi-filter ensemble method for gene selection," *Genes (Basel).*, vol. 9, no. 5, p. 258, 2018.

[15] A. Rau, M. Flister, H. Rui, and P. L. Auer, "Exploring drivers of gene expression in the Cancer Genome Atlas," *Bioinformatics*, vol. 35, no. 1, pp. 62–68, 2019.

[16] I.-S. Jeong, H.-K. Kim, T.-H. Kim, D. H. Lee, K. J. Kim, and S.-H. Kang, "A feature selection approach based on simulated annealing for detecting various denial of service attacks," *Softw. Netw.*, vol. 2018, no. 1, pp. 173–190, 2018.

[17] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, 2012.

[18] Y.-L. Wu, C.-Y. Tang, M.-K. Hor, and P.-F. Wu, "Feature selection using genetic algorithm and cluster validation," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2727–2732, 2011.

[19] M. M. Kabir, M. Shahjahan, and K. Murase, "A new hybrid ant colony optimization algorithm for feature selection," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3747–3763, 2012.

[20] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *ournal R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[21] A. Al Mamun and A. M. Mondal, "Feature Selection and Classification Reveal Key lncRNAs for Multiple Cancers," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (IEEE BIBM)*, 2019, pp. 2825–2831.

[22] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1–3, pp. 389–422, 2002, doi: 10.1023/A:1012487302797.

[23] J. Fang, "Tightly integrated genomic and epigenomic data mining using tensor decomposition," *Bioinformatics*, vol. 35, no. 1, pp. 112–118, 2019.

[24] M. B. Pouyan and D. Kostka, "Random forest based similarity learning for single cell RNA sequencing data," *Bioinformatics*, vol. 34, no. 13, pp. i79--i88, 2018.

[25] M. Ram, A. Najafi, and M. T. Shakeri, "Classification and biomarker genes selection for cancer gene expression data using random forest," *Iran. J. Pathol.*, vol. 12, no. 4, p. 339, 2017.

[26] R. Wang, "AdaBoost for feature selection, classification and its relation with SVM, a review," *Phys. Procedia*, vol. 25, pp. 800–807, 2012.

[27] T. T. Le, R. J. Urbanowicz, J. H. Moore, and B. A. McKinney, "Statistical inference Relief (STIR) feature selection," *Bioinformatics*, vol. 35, no. 8, pp. 1358–1365, 2019.

[28] X. Lu, H. Gu, Y. Wang, J. Wang, and P. Qin, "Autoencoder based feature selection method for classification of anticancer drug response," *Front. Genet.*, vol. 10, p. 233, 2019.

[29] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *J. Educ. Psychol.*, vol. 24, no. 6, p. 417, 1933.

[30] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (80-.).*, vol. 313, no. 5786, pp. 504–507, 2006.

[31] A. Mirzaei, V. Pourahmadi, M. Soltani, and H. Sheikhzadeh, "Deep feature selection using a teacher-student network," *Neurocomputing*, vol. 383, pp. 396–408, 2020.

[32] Y. Lu, Y. Fan, J. Lv, and W. S. Noble, "DeepPINK: reproducible feature selection in deep neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 8676–8686.

[33] V. Borisov, J. Haug, and G. Kasneci, "CancelOut: A Layer for Feature Selection in Deep Neural Networks," in *International Conference on Artificial Neural Networks*, 2019, pp. 72–83.

[34] A. Abid, M. F. Balin, and J. Zou, "Concrete autoencoders for differentiable feature selection and reconstruction," *arXiv Prepr. arXiv1901.09346*, 2019.

[35] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv Prepr. arXiv1611.00712*, 2016.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv Prepr. arXiv1312.6114*, 2013.

[37] M. Goldman, B. Craft, A. Brooks, J. Zhu, and D. Haussler, "The UCSC Xena Platform for cancer genomics data visualization and interpretation," *BioRxiv*, p. 326470, 2018.

[38] O. Chapelle and S. S. Keerthi, "Multi-class feature selection with support vector machines," in *Proceedings of the American statistical association*, 2008, vol. 58.

[39] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognit. Lett.*, vol. 31, no. 14, pp. 2225–2236, 2010.

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

[41] Mart\'\in~Abadi *et al.*, "{TensorFlow}: Large-Scale Machine Learning on Heterogeneous Systems." 2015.

[42] F. Chollet and others, "Keras." GitHub, 2015.

[43] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008, Accessed: Sep. 03, 2019. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[44] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *J. Am. Stat. Assoc.*, vol. 53, no. 282, pp. 457–481, 1958.

[45] E. A. Mauger, R. A. Wolfe, and F. K. Port, "Transient effects in the cox proportional hazards regression model," *Stat. Med.*, vol. 14, no. 14, pp. 1553–1565, 1995, doi: 10.1002/sim.4780141406.

[46] A. Al Mamun and A. M. Mondal, "Long Non-coding RNA Based Cancer Classification using Deep Neural Networks," in *10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (ACM BCB'19)*, 2019, pp. 541–541, Accessed: Nov. 10, 2019. [Online]. Available: https://dl.acm.org/citation.cfm?id=3343249.