scientific reports

Check for updates

OPEN

Forecasting influenza hemagglutinin mutations through the lens of anomaly detection

Ali Garjani[1], Atoosa Malemir Chegini[1], Mohammadreza Salehi[1], Alireza Tabibzadeh[2], Parastoo Yousefi[2], Mohammad Hossein Razizadeh[2], Moein Esghaei[3], Maryam Esghaei[2] & Mohammad Hossein Rohban[1]✉

The influenza virus hemagglutinin is an important part of the virus attachment to the host cells. The hemagglutinin proteins are one of the genetic regions of the virus with a high potential for mutations. Due to the importance of predicting mutations in producing effective and low-cost vaccines, solutions that attempt to approach this problem have recently gained significant attention. A historical record of mutations has been used to train predictive models in such solutions. However, the imbalance between mutations and preserved proteins is a big challenge for the development of such models that need to be addressed. Here, we propose to tackle this challenge through anomaly detection (AD). AD is a well-established field in Machine Learning (ML) that tries to distinguish unseen anomalies from normal patterns using only normal training samples. By considering mutations as anomalous behavior, we could benefit existing rich solutions in this field that have emerged recently. Such methods also fit the problem setup of extreme imbalance between the number of unmutated vs. mutated training samples. Motivated by this formulation, our method tries to find a compact representation for unmutated samples while forcing anomalies to be separated from the normal ones. This helps the model to learn a shared unique representation between normal training samples as much as possible, which improves the discernibility and detectability of mutated samples from the unmutated ones at the test time. We conduct a large number of experiments on four publicly available datasets, consisting of three different hemagglutinin protein datasets, and one SARS-CoV-2 dataset, and show the effectiveness of our method through different standard criteria.

The influenza virus infection is mostly presented as a self-limited respiratory infection in immunocompetent people. However, influenza viruses could lead to a life-threatening infection in the elderly and other risk group patients. Hemagglutinin is a glycoprotein located on the surface of influenza viruses and acts as an attaching ligand to the host cells and inserts the virus into the cells. To escape from immune responses, the virus can alter the antigenic features of the hemagglutinin (HA) protein by point mutations. This phenomenon is known as antigenic drifts. Mutations in the genes of influenza viruses could cause antigenic drift by changing the HA protein structure[1–4]. This results in a new strain of the virus that is not effectively recognized by the immune system and making the virus spread easily and causing an epidemic. Influenza viruses are classified in the *Orthomyxoviridae* family. In this family, there are three important human pathogens, including *Alphainfluenzavirus*, *Betainfluenzavirus*, and *Gammainfluenzavirus*. Influenza A virus is the only member of *Alphainfluenzavirus*, and is an important human pathogen due to its wide host range and the higher rate of drift and shift mutations[5–7].

HA is the main part of the virus attachment to the host cell receptor. The globular head domain of HA, which is critical for neutralizing antibody generation by the host immune system, is one of the most potent genomic locations for mutation. Influenza A viruses are divided into two different groups based on the globular head domain. The HA 1, 2, 5, 6, 9, 11, 12, 13, 16, and 18 types are placed in one group, and types 3, 4, 7, 10, 14, and 15 are considered members of the second group of HAs[8]. The Cb, Ca, Sb, and Sa are four important antigenic sites in the H1 domain of HA[9–11]. The amino acid residues number 143, 156, 158, 190, 193, and 197 are the most

[1]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. [2]Department of Virology, School of Medicine, Iran University of Medical Sciences, Tehran, Iran. [3]Cognitive Neuroscience Laboratory, German Primate Center, Leibniz Institute for Primate Research, Goettingen, Germany. ✉email: rohban@sharif.edu

important residues for evolutionary and antigenic features of HA1[12]. The role of HA1 mutations in the adequacy of influenza vaccination has made WHO Collaborating Centers and Vaccines and Related Biological Products Advisory Committee (VRBPAC)[13] responsible for functional monitoring, reports, and decision for new season vaccines. Despite this delicate process, there are shortages and some strain mismatches between the vaccine strains and circulating strains[14,15].

In the current study, we also evaluated the SARS-CoV-2 Spike mutations as an extra evaluation and a demo for future consideration in this field. The SARS-CoV-2 (severe acute respiratory syndrome coronavirus 2) is the etiological agent for the COVID-19 (Coronavirus disease-2019) pandemic. The SARS-CoV-2 is a Betacoronavirus and a member of the sarbecoviruses sublineage[16]. The virus genome is an ssRNA (Single strand RNA) of 34kb in length. SARS-CoV-2 contains different genes including ORF1a/b, Spike (S), Envelope (E), Membrane (M), Nucleoprotein (N), and accessory ORFs[17]. The virus binds into cells by using the S protein attachment to the cellular receptor ACE-2 (Angiotensin-converting enzyme 2)[18]. The S protein is the most important antigenic part of the virus[19].

In recent years, Artificial Intelligence (AI) algorithms have achieved human or even super-human performance on different tasks such as image classification[20], text classification[21], action recognition[22], etc. Anomaly Detection (AD) is a sub-domain of AI that is responsible for learning a normal representation space and detecting anomalous samples at the test time by exploiting the learned representation. Due to different challenges in the labeling of anomalous samples, such as the high cost or rareness of such samples, most methods in this domain only use normal samples for the training. This is called unsupervised AD. Alternatively, one may use a very limited number of labeled anomalous samples in the training process, which is called the semi-supervised AD[23].

Unsupervised[24–28] and semi-supervised[29,30] anomaly detection methods have recently achieved satisfactory results on a variety of domains such as image, text, time-series, and video. Deep Semi-Supervised Anomaly Detection (DeepSAD)[29], as a recently proposed semi-supervised AD method, made clear that semi-supervised anomaly detectors are significantly superior compared to the supervised training classification algorithms, specifically when the training dataset is complex, and the number of normal samples is much higher than the anomalous ones. This is because anomaly detectors attempt to find a compact representation space for the normal samples while maximizing the margin that exists between normal and abnormal ones. This helps them to learn the most general and unique features of the normal samples, and not rely overly on the contrast that exists between normal and anomalous samples to classify them.

Since in the mutation prediction tasks the number of unmutated samples is much higher than the mutated ones, the problem can be formulated as an anomaly detection task. In this formulation, unmutated and mutated samples are considered as normal and anomalous samples, respectively. The benefits of this approach are twofold. Firstly, a semantically meaningful representation could be learned even with a small number of training samples, which makes generalization to unseen test time samples possible. Secondly, as the finding and labeling procedure of mutated viruses is an expensive and time-consuming process, anomaly detectors could work fine with, or without a limited number of anomalous or mutated, training samples[23].

By this motivation, we propose the first anomaly detection framework for predicting virus mutations. We use the Long Short-Term Memory (LSTM)[31] neural network in combination with the Deep Semi-Supervised Anomaly Detection (DeepSAD) loss[29] to not only learn long-term input dependencies but also to find a semantic representation space for the mutated and unmutated training samples. Figure 1 shows the overall architecture of the proposed method. We conduct extensive experiments to show the effectiveness of our method in improving the average recall, F1-score, precision, and Area Under the Curve (AUC) for three different publicly available Influenza datasets.

Background

For the sake of clarity, we discuss some of the important prerequisites from deep learning literature in this section. At first, some Recurrent Neural Network architectures, such as LSTMs[31], are discussed. Then, a brief introduction to the anomaly detection methods is presented.

Recurrent neural networks (RNN). RNNs are broadly used to model the data sequential dependencies, where the sequence could be formed based on temporal or spatial arrangements. Initial architectures of RNNs, such as the vanilla RNN, suffer from memorizing long-term as well as short-term dependencies. To address this issue, alternative architectures, such as LSTM[31] networks, bi-directional RNNs[32], and gated recurrent units[33] GRU's have been introduced. All these approaches attempt to summarize previous inputs into their hidden state that is updated in each time step t. The mentioned information is regulated using some parameters or gates. For instance, the LSTM network consists of LSTM cells. Each cell contains a state, h_t, and memory, s_t. These two are updated based on three different gates that are called *input gate*, i_t, *forget gate*, f_t, and *output gate*, o_t. The input gate selects some of the memory dimensions to modify (Eq. 2). The forget gate decides which memory cell dimensions should be ignored in the next time step (Eq. 1). The output gate decides which dimensions of the memory should be transferred to the state (Eq. 3). The cell and state vectors are updated based on these gates and activation values that are produced through the tanh activation (Eqs. 4, 5). Specifically, the memory constitutes previous memory dimensions that are not forgotten, plus the input activation values that the input gate selects. Finally, the state constitutes memory activation values that are selected by the output gate. In all equations the parameters W_f, W_i, W_o, W_s and b_f, b_i, b_o, b_s are shared between the cells and are learned during the training process. This recurrent behavior of the LSMT model, where h_t and s_t are a function of h_{t-1} and s_{t-1}, these models have the ability to take input from varying lengths, and the length of the sequence does not need to be fixed.

Note that a sigmoid activation function is used in the gates to map the gate outputs between zero and one, which models the selection, i.e., gate output of 1 represents the complete selection of an embedding, and the 0

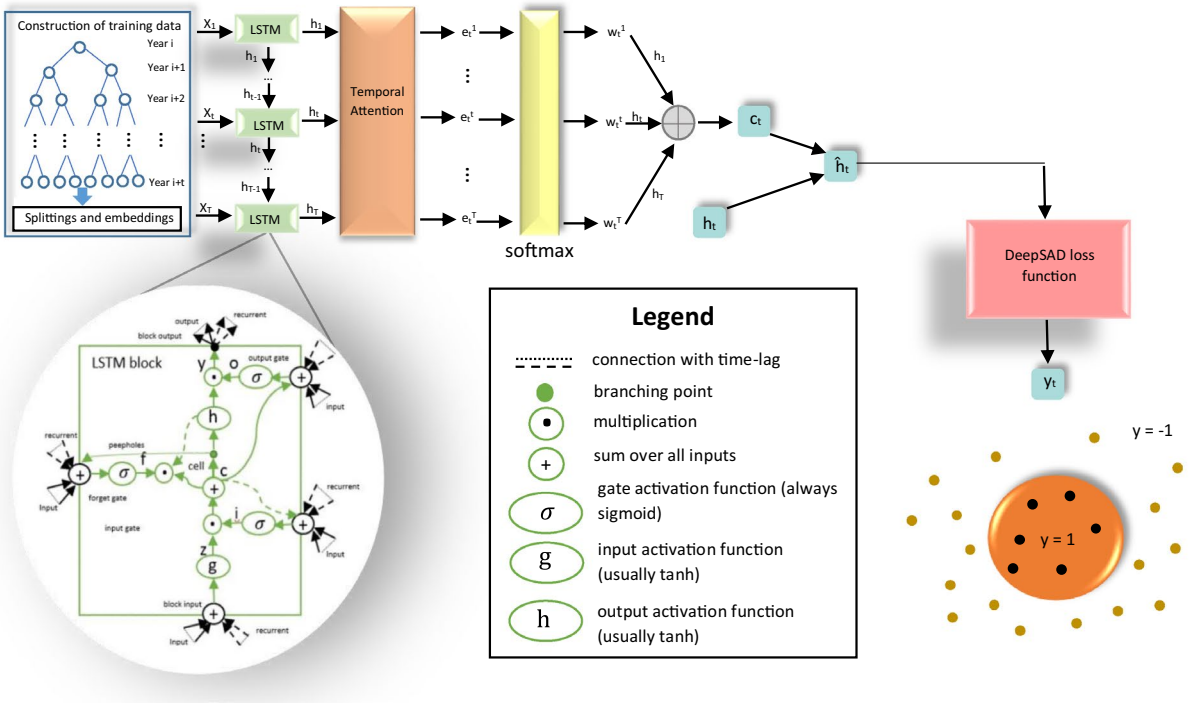

Figure 1. The overall architecture of our method. First, the raw data is processed and the output $(X_1^t, X_2^t, \ldots, X_n^t)$ is prepared at the time step t, where n is the embedding dimensions, t denotes the time. After the pre-processing phase, LSTM cells are used to produce hidden states, h_i, for each time point t. Then, the attention function takes h_i and the cell state s_{t-1}, and outputs e_t^i. Next, by using a softmax function, the weights w_t^i's are produced. The weighted sum of the hidden states, h_i's, is obtained by using the mentioned weights. The output of this weighted sum, c_t, and the hidden state h_t will then be used to produce the encoded vector $\hat{h}_t$. At the last step, the DeepSAD loss function is applied to $\hat{h}_t$ to decide whether the input data is in-class (normal) 1 or out-class (anomaly) − 1.

value corresponds to a complete non-selection. A tanh activation function is used in the cell and state update rules to produce activation values that are between − 1 and 1.

$$f_t = \sigma(W_f[h_{t-1}; x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i[h_{t-1}; x_t] + b_i) \tag{2}$$

$$o_t = \sigma(W_o[h_{t-1}; x_t] + b_o) \tag{3}$$

$$s_t = f_t \odot s_{t-1} + i_t \odot tanh(W_s[h_{t-1}; x_t] + b_s) \tag{4}$$

$$h_t = o_t \odot tanh(s_t) \tag{5}$$

Despite huge efforts on making different LSTM architectures to improve its performance, no architecture has been proposed yet that is generally better than the original one[34]. Therefore, our proposed method is based on the LSTM networks with some improvements on its ability to maintain long-term information and interpretability.

Anomaly detection. As mentioned before, anomaly detection is a sub-branch of artificial intelligence seeking to solve one-class classification problems. One-class methods only access the labels of one category of a dataset, called the normal class[35]. These methods then seek to design a classifier that can distinguish the normal class vs. the unseen classes, which is also referred to as anomaly classes. For instance, in mutation prediction problems, the anomaly detection method assumes access to only unmutated samples. This setup could be adopted for reasons such as the large cost of the data-gathering process from both kinds of mutated and unmutated classes, or even the impossibility of gathering all kinds of mutations in our training dataset. Such issues make the classification setup ineffective, as the classifier may get biased towards accurate prediction of only known mutations that are reflected in the training set. Deep Support Vector Data Description (DSVDD)[24] is one of the basic anomaly detection methods that is trained in an unsupervised manner. It tries to find a latent space and the most compact hyper-sphere that contains the normal training samples in this space. The pre-assumption of DSVDD is that anomalous samples layout of the circle in contrast to normal ones, which could make them detectable. Recently, Chong et al.[36] have shown the vulnerability of this method to the mode collapse, i.e., convergence of all

data points to a single point in the latent space, due to its unsupervised training process. To alleviate this issue, DSAD[29] suggests using a limited number of anomalous training samples to train DSVDD and achieved satisfying results. It has shown that a limited number of anomalous samples is enough to not only prevent the mode collapse but also enhance the supervised classifiers' accuracy, specifically when training samples are significantly imbalanced toward the normal class.

From a different point of view, autoencoder(AE) is another dominant framework that is used in the field. Owing to their unsupervised training process, they are intrigued for formulating anomaly detection problems based on their abilities. They are trained on normal training samples by this pre-assumption that they would be reconstructed well at the test time compared to the abnormal inputs. As the primary versions of autoencoders have shown deficiencies in their performance when the training dataset becomes complex, some variants of AE-based methods have been proposed based on generative adversarial networks [37–39], adversarial robust training [25], or self-supervised learning methods [40].

Experiments and results

Dataset. For the experiments on HA, we used the dataset provided in Tempel[41]. The data is hosted at https://drive.google.com/drive/folders/1-pJGBsVfIqCEizetTQe43OQJvkmhocdW. This dataset includes influenza subtypes H1N1, H3N2, and H5N1, which have sequence lengths of 566, 566, and 568, respectively. The number of available HA sequences in each year for all three subtypes is provided in Table 1.

For the experiments on SARS-CoV-2, we selected the RBD domain of spike nucleotide sequences, with a length of 1273, in the United Kingdom from January 1, 2020, to the last day of December 2020. The number of the available sequences in each month is provided in Table 2. In our current study, we just used one year period of time for SARS-CoV-2 as an introduction for future research and the data and results seem promising but limited and not completely validated. This mutation prediction for the SARS-CoV-2 Spike gene needs future studies

The studies are performed in accordance with the Declaration of Helsinki and are carried out in accordance with the relevant guidelines and regulations.

Implementation. We use PyTorch and Scikit-learn in our implementations. For the experiments that are performed on H1N1, H3N2, and H5N1, similar to the previously proposed method Tempel[41], the first 1000 samples per year are chosen for training and testing. Then, the first 80% of the samples in each year are selected as the training set, and the remaining 20% are set as the test set. For our model, since validation is also needed for finding the best threshold, the first 10% of the training set is used for validation, and the remaining 90% for training. We use a batch size of 256, a learning rate of 0.001, and gradient descent for the optimization. Also, similar to Tempel, hidden layer size and dropout percentage are set to 128 and 0.5, respectively. As Fig. S3 shows, validation curves can be used as a good hint to stop the training process on the 50th epoch, in which the model has converged. Therefore, we have trained our model for 50 training epochs.

The bioinformatics pipeline for the suggested amino acid alteration locations. All of the amino acid alteration locations that are suggested by the algorithm are evaluated by a simple alignment analysis. After the model training, possible amino acid alteration locations for 2016 influenza H1N1, H3N2, and H5N1 are listed based on the highest recall and precision. The influenza virus sequences for 2016 are obtained from the NCBI influenza database. A random 100 full-length samples from 2016 influenza circulating strains are used as a sample for amino acid alteration locations evaluation. In addition, the alterations are evaluated based on the suggested vaccine strains for the 2016–2017 season (stains include: A/California/7/2009 (H1N1)pdm09-like virus, A/Hong Kong/4801/2014 (H3N2)-like virus and B/Brisbane/60/2008-like virus (B/Victoria lineage))[42]. All of the sampling influenza sequences from 2016 are aligned, and the alignment and amino acid locations

Year	2001	2002	2003	2004	2005	2006	2007	2008
H1N1	77	40	77	60	92	139	307	322
H3N2	53	135	215	180	204	132	285	189
H5N1	21	26	45	117	169	231	291	171
Year	2009	2010	2011	2012	2013	2014	2015	2016
H1N1	2517	957	794	586	715	499	391	687
H3N2	340	411	577	868	704	876	1036	932
H5N1	150	169	188	134	166	143	135	17

Table 1. Number of HA samples for each subtypes from 2001 to 2016.

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
SARS-CoV-2	3	116	900	1500	1000	960	986	1031	800	1146	1416	900

Table 2. Number of sequence samples from January 2020 to December 2020.

are visualized in CLC Workbench[43]. Major epitopes are marked based on the previous studies for H1N1[44] and H3N2[45]. The H5N1 influenza was not reported for the human host during 2016, hence for the mutation assessment for this dataset, we use all of the hosts during 2016 and mostly avians.

Baselines. In this section, we compare our method with other recently proposed anomaly detection (AD) methods that can be easily adapted to our task. Although some approaches such as Golan et al.[46] and Bergman et al.[47] achieve top performance in anomaly detection problems, they use self-supervised learning methods that are specialized for the image processing tasks. Consequently, they reach weak performance on our datasets. Table 3 includes the performance of Bergman et al., called GOAD, compared to the other methods. The parameter T represents the time-series sequence length ending in the year 2016. For the sake of equality in our comparisons, we have chosen T to be 5, 10, and 15, similar to Tempel[41]. Besides, repeating experiments for different T values gives a more comprehensive intuition on the sensitivity of the methods.

Some AD algorithms are not domain-specific such as autoencoder-based (AE) approaches. We have reported the performance of ARAE[25] as a stable, domain-agnostic, and high-performance approach. As ARAE is trained in a fully unsupervised training manner, using only unmutated training samples, its results are not competitive with the semi-supervised learning methods. Besides, AEs are not effective when facing complex datasets.

Moreover, we report the performance of a similar semi-supervised anomaly detection method that has been proposed recently, ESAD[48]. It uses an AE-Based approach but employs both kinds of negative (unmutated) and positive (mutated) samples in its loss function. Finally, the performance of the original vanilla DeepSAD method, without our proposed modifications in Eq. (12), is reported in Table 3.

Results. We report the mean and standard deviation of the metrics that are used to evaluate our method in different experiments. The model is trained for 5 trials. Tables 4, 5, 6, and 7 show the performance of our method compared to Tempel[41], which is a recently proposed SOTA on the HA datasets. For each experiment, Area Under Curve (AUC), F1-Score, recall, and precision are reported as in other related works. To report Tempel's results, we use the original implementation that is publicly available on the reported link in their paper. Following the experiments on HA, we have also conducted similar experiments on the SARS-CoV-2 dataset for $T \in \{5, 10\}$. Unlike the experiments on HA, for these experiments, T presents time-series sequence length in months rather than years.

Dataset	Model	Precision			Recall			F1-score		
		T = 5	T = 10	T = 15	T = 5	T = 10	T = 15	T = 5	T = 10	T = 15
H1N1	ARAE	0.162	0.171	0.181	0.472	0.501	0.532	0.241	0.255	0.270
	GOAD	0.124	0.132	0.190	0.361	0.391	0.555	0.185	0.197	0.283
	ESAD	0.384	0.406	0.419	0.570	0.573	0.589	0.459	0.475	0.490
	DeepSAD	0.294	0.318	0.322	0.497	0.477	0.478	0.369	0.382	0.385
H3N2	ARAE	0.171	0.210	0.219	0.498	0.611	0.639	0.255	0.313	0.326
	GOAD	0.152	0.178	0.190	0.444	0.568	0.555	0.226	0.271	0.283
	ESAD	0.279	0.290	0.331	0.499	0.511	0.509	0.358	0.370	0.401
	DeepSAD	0.343	0.359	0.392	0.536	0.540	0.565	0.418	0.431	0.463
H5N1	ARAE	0.175	0.187	0.186	0.488	0.505	0.512	0.258	0.273	0.273
	GOAD	0.217	0.238	0.235	0.538	0.566	0.569	0.309	0.335	0.333
	ESAD	0.347	0.371	0.350	0.568	0.538	0.482	0.431	0.439	0.406
	DeepSAD	0.418	0.453	0.471	0.620	0.617	0.619	0.499	0.522	0.535

Table 3. Precision, recall and F1-score on H1N1, H3N2 and H5N1 datasets for $T \in \{5, 10, 15\}$. Results are for four different methods including ARAE, GOAD, ESAD and original DeepSAD.

Dataset	Method	T = 5	T = 10	T = 15	Mean
H1N1	Tempel[41]	0.8467	0.8537	0.8455	0.8486
	Ours	**0.9713 ± 0.0002**	**0.9713 ±0.00008**	**0.9715 ± 0.00015**	**0.9713**
H3N2	Tempel[41]	0.8989	0.8863	0.8884	0.8912
	Ours	**0.9419 ± 0.00004**	**0.9419 ± 0.00004**	**0.9416 ± 0.0006**	**0.9418**
H5N1	Tempel[41]	0.9657	0.9696	0.9671	0.9674
	Ours	**0.9829 ± 0.00017**	**0.983 ± 0.00021**	**0.9819 ± 0.00026**	**0.9826**

Table 4. AUC comparison on H1N1, H3N2 and H5N1 datasets for $T \in \{5, 10, 15\}$. Mean is the average of the results on T for each dataset and model. Our results are significantly better or competative with the SOTA method. Significant values are in bold.

Dataset	Method	T=5	T=10	T=15	Mean
H1N1	Tempel[41]	**0.8212**	**0.821**	**0.8213**	**0.8211**
	Ours	0.8178 ± 0.0006	0.8176 ± 0.0004	0.81992 ± 0.0004	0.8184
H3N2	Tempel[41]	0.5957	0.5785	0.5913	0.5885
	Ours	**0.5966 ± 0.0009**	**0.5979 ± 0.003**	**0.60127 ± 0.001**	**0.5986**
H5N1	Tempel[41]	0.5167	0.5287	0.51722	0.5208
	Ours	**0.53083 ± 0.003**	**0.5387 ± 0.001**	**0.5287 ± 0.01**	**0.5327**

Table 5. F1-Score of predictions on H1N1, H3N2 and H5N1 datasets for $T \in \{5, 10, 15\}$. Mean is the average of the results on T for each dataset and model. Significant values are in bold.

Dataset	Method	T = 5	T = 10	T = 15	Mean
H1N1	Tempel[41]	0.8012	0.7997	0.8013	0.8007
	Ours	**0.8033 ± 0.003**	**0.8066 ± 0.003**	**0.8071 ± 0.006**	**0.8056**
H3N2	Tempel[41]	0.5106	0.4793	0.4943	0.4947
	Ours	**0.6107 ± 0.01**	**0.63305 ± 0.01**	**0.62768 ± 0.02**	**0.6238**
H5N1	Tempel[41]	0.3671	0.38181	0.36643	0.3717
	Ours	**0.4158 ± 0.01**	**0.4205 ± 0.009**	**0.42146 ± 0.01**	**0.4192**

Table 6. Recall of predictions on H1N1, H3N2 and H5N1 datasets for $T \in \{5, 10, 15\}$. Mean is the average of the results on T for each dataset and model. Our results are competitive with SOTA on these datasets. Significant values are in bold.

Dataset	Method	T = 5	T = 10	T = 15	Mean
H1N1	Tempel[41]	**0.8423**	**0.8445**	**0.8389**	**0.8419**
	Ours	0.8330 ± 0.004	0.8292 ± 0.004	0.8333 ± 0.006	0.8318
H3N2	Tempel[41]	**0.7147**	**0.7300**	**0.7358**	**0.7268**
	Ours	0.5845 ± 0.0004	0.5694 ± 0.01	0.5786 ± 0.01	0.5775
H5N1	Tempel[41]	**0.8721**	**0.8598**	**0.8791**	**0.7803**
	Ours	0.7530 ± 0.02	0.7606 ± 0.02	0.7176 ± 0.02	0.7437

Table 7. Precision of predictions on H1N1, H3N2 and H5N1 datasets for $T \in \{5, 10, 15\}$. Mean is the average of the results on T for each dataset and model. Significant values are in bold.

Comparison of AUC, F1-score, recall, and precision. As it is shown, our method achieves competitive or significantly superior results in all standard criteria AUC, F1, recall, and precision averaged across all T values on HA datasets. It is good to notice that our method is consistently better than Temple when the parameter T is small, i.e. $T = 5$, on HA training, sets for all different measures. This could be the effect of using an anomaly detection loss to train our model. Since DeepSAD loss attempts to find a shared representation space for the unmutated samples, it uses the given training set efficiently and, in the best scenario, extracts the most general features of them.

Stopping criterion of the training process. As mentioned before, we use validation data to find the best point to cut the training process. Figure S3 shows the F1-score curves of the validation and test tests. As is depicted, the validation curves almost always follow their corresponding test curves, which shows their usability in determining the points to stop the training process. Note that the smaller the value of parameter T, the more the complexity of finding a good representation space. Therefore, a larger number of training samples is needed to approximately make the validation and test distributions similar, and consequently, some validation curves do not completely follow the test curves.

Comparison of ROC curves. Figures S4 and S5 show the ROC curves of our methods compared to Tempel. As it is obvious, our method always has a lesser value of false-positive rate for high true positive rate values for all T values on all HA test datasets. This can be justified by the margin that exists between the normal and abnormal distributions. The HSC loss attempts to make this margin as large as possible, but classification-based approaches such as Tempel only try to minimize the cross-entropy loss that does not consider maximizing the margin explicitly. This helps our model to be more robust against noises that exist in our training datasets, which

are typical considering the difficulties of the dataset-making process. Moreover, for the SARS-CoV-2 dataset, our method, in setups with lower false-positive rates, shows a higher true-positive rate compared to Tempel.

Ablation study results. We have also conducted further experiments using a different model called transformers for predicting the mutations. The results for these experiments on HA are shown in Table 8. Details about the mechanism of the transformers are provided in the ablation section of methods.

Mutation prediction results in the real-world dataset assessment. After the evaluation of possible amino acid alterations, mutations that exhibit top recall and precision rates for T = 5, T = 10, and T = 15 are listed and assessed based on random samples of the 2016 reported influenza sequences and the 2016–2017 season vaccine strain. Table S1 represents the model's results for different possible amino acid alteration locations and their presence in the random sampling data from the 2016 reported stains. It should be noted that based on the model's results, each reported alteration, as an amino acid number is associated with its previous and next amino acid. The proposed alterations by the algorithm in sampling data are evaluated separately and regardless of the algorithm ROC curve for the mutations. Furthermore, the proposed alteration positions that do not depict any mutations in Figs. S1 and S2 or Table S1 in the sampling aligned data from the 2016 year do not express that there is no mutation in all of the years of the data set for that particular position. The mutations in these particular locations do not appear in the current random sampling due to the low stability or frequency of the mutation.

H1N1 influenza A. By the evaluation of the antigenic sites of influenza H1N1 amino acid sequence, the proposed model highlights some important alterations in amino acid numbers 168, 170, 205, and 253 in three important sites Ca, Sa, and Sb. Meanwhile, there are too many missed important amino acid alterations in Sa, Sb, and especially in Ca. More details are provided in Fig. S1. In the comparison of our current anomaly detection method with the suggested method by Tempel, there are three differences in alteration positions. These differences refer to the amino acid numbers 165 and 174 by Tempel and 166 in our model. The residues 165 and 166 did not represent any important alterations while 174 is located in Sa.

H3N2 influenza A. By the evaluation of the antigenic sites of influenza H3N2 amino acid sequence, the model proposed some important alterations in amino acid number 192 in antigenic Site B and amino acid 226 in the Receptor binding site (RBS). Some alterations in other locations are illustrated in Fig. S2. It should be mentioned that some important alterations in antigenic sites Lower HA, site A, and Site B are missed. In the comparison of our model and Tempel, each model proposed two positions that the other model did not detect. The residues 88 and 260 are suggested by the Tempel model and not by our current model. Residue 88 represents variable positions in the sequence but not epitope. While amino acid number 260 is an epitope location and is missed by our model. Our model suggests an alteration in the amino acids numbers 177 and 193 more than the Tempel model. The amino acid 177 does not reflect any epitope or high prevalence variation, while the 193 is a true mutation in site B.

H5N1 influenza A. The influenza H5N1 amino acid sequence was not evaluated in the reported sequences from the 2016 human host due to the lack of reports in that particular year. For the evaluation of the model and multiple sequence alignment for more important alteration positions in Table S1, we used the avian sequences of the H5N1 during 2016 due to the nature of the virus. The amino acids numbers 82, 116, 152, 166, 172, 179, 185, 205, and 242 seem to be important for further studies and antigenic evaluation. Moreover, by comparing our current model and the Tempel model, there are two different mutation locations in each (118 and 130 for Tempel in comparison with 172 and 179). The conducted study reveals two important mutation positions in SARS-CoV-2. The data was collected on 8 June 2021. The proposed mutation positions are the 476 and 500 amino acid locations. The amino acid number 500 represents the Alpha variant-specific mutation (N501Y) on the date of the dataset preparation. Another proposed mutation position in 476 did not represent any particular association with any variants except the Omicron variant (S477N). It has to be noted that at the time of the database preparation, there was no clue about the Omicron variant, and this mutation in the 476 positions did not reflect any particular result.

Dataset	Method	Recall			F1-score			AUC		
		5	10	15	5	10	15	5	10	15
H1N1	Tempel	0.80128	0.79972	0.80137	0.82128	0.82102	**0.82138**	0.84677	0.85376	0.84551
	Transformer	**0.8024**	**0.80137**	**0.80432**	**0.82151**	**0.8218**	0.81867	**0.9717**	**0.9711**	**0.96935**
H3N2	Tempel	0.51069	0.479381	0.49432	0.59571	0.57859	**0.59137**	0.89896	0.88632	0.88847
	Transformer	**0.63373**	**0.632757**	**0.5949**	**0.6196**	**0.61002**	0.5902	**0.9445**	**0.9354**	**0.91917**
H5N1	Tempel	0.3671	0.38181	0.36643	0.5167	**0.52878**	**0.51722**	0.96572	0.9696	0.96715
	Transformer	**0.38083**	**0.43426**	**0.42167**	**0.5174**	0.48174	0.49658	**0.98874**	**0.97638**	**0.97603**

Table 8. Recall, F1-score and AUC on H1N1, H3N2 and H5N1 datasets for $T \in \{5, 10, 15\}$. Results are for The Transformer Network and Tempel. Significant values are in bold.

Discussion

Every year, hundreds of thousands of deaths are reported from influenza disease. Despite many efforts to develop new treatments and vaccines, due to the high genetic diversity of the influenza viruses (IVs), complete success has not been achieved yet. This extended genetic diversity is due to the RNA genome of these viruses. Between the proteins encoded by the IVs genome, hemagglutinin (HA) is among the most important proteins that play key roles in the infectivity and propagation of the virus[49]. This glycoprotein plays a paramount role in binding to the Sialic acid, the ligand of the virus at the surface of the host cell, as well as fusion into the cell. In fact, the pathogenicity of IVs depends on efficient cleavage of hemagglutinin precursor to HA1 and HA2 proteins, in which the former is responsible for receptor-binding activity and the latter anchors the HA1 and is also responsible for pH-dependent fusion[50]. This reason, along with the increased resistance to currently available classes of drugs, which inhibit M2 ion channels and neuraminidase, has made inhibition of hemagglutinin one of the attractive goals for the development of anti-influenza drugs in recent years[51]. In addition to the medications, the immune system uses HA as a target in response to infection. Once the infection is established, the adaptive immune system triggers a strong response against the virus, in which neutralizing antibodies are produced against the virus. HA is the major target of these neutralizing antibodies[52]. Since these antibodies are strain specific[53], mutations in the HA epitopes that are targeted by these antibodies may change the antigenicity of the virus and lead to the emergence of antibody- and vaccine-escape strains[52,54]. After the entrance of viruses to the body and the production of neutralizing antibodies by the immune cells, antibody-escape viruses evade the host immune system, a phenomenon called the selection of "antibody escape" variants[52].

The influenza virus is highly prone to antigenic changes. These changes are often caused by two main mechanisms called antigen shift and drift. If a cell is infected with two different genotypes of the IVs, a new strain may develop due to the placement of different parts of the two strains into new viral particles. This phenomenon, which can lead to the generation of new pandemic strains, is called the antigen shift. Influenza viruses are highly prone to point mutations due to the lack of proofreading ability of their RNA genome. The accumulation of these point mutations is called antigen drift. Occurrence of this phenomenon in the gene encoding HA causes alterations in the structure and function of this protein[55]. As a result, the immune system is no longer able to detect the virus[56]. Therefore, this jeopardizes us at risk of future pandemics to which our bodies have no resistance[57].

Since HA has a critical role in the replication cycle of IVs, the occurrence of these changes has greatly contributed to the evolution of these viruses[58]. A series of mutations with few effects on the antigenicity of the IVs finally lead to intense antigenic drift, a phenomenon called "cluster transition," which is the characteristic of the evolution of IVs[59]. Studies have shown that changes in the amino acid sequence of HA1 can tremendously alter the antigenicity of these viruses[56]. Thereby tracking these changes is essential to predict the future behaviors of the virus. Thyagarajan and Bloom mutagenized the HA gene of wild-type IVs to create codon-mutant libraries of the HA gene and then used these libraries to make a pool of mutants by reverse genetics. Using the Illumina deep mutational scanning, they found that there are more than 10,000 different probabilities for mutation in this protein. They have also cultured these mutants and investigated the mutated viruses. They have concluded that mutations mostly occur in the regions of the HA protein that is recognized by antibodies. Otherwise, the receptor-binding domain is less frequently subject to mutations and it is the possible reason why HA still targets the Sialic acid in spite of extensive mutations in this protein[60].

Determining the new antigenic variants of the IVs is critical in order to develop efficient flu vaccines. The WHO collaborates with a number of laboratories around the world to identify circulating influenza viruses in the human population ([61]. Traditionally, the hemagglutination inhibition (HI) test has been used to evaluate antigenic variants of the virus, which is a time-consuming and hand-operated method[62]. However, advances in computer science and bioinformatics have led to the invention and development of faster and more accurate methods. By using scoring and regression methods on the sequences collected from the H3N2 flu virus between 1971 and 2002, Liao et al. proposed a method for predicting variants of the virus. According to their proposed method, influenza virus variants between 1999 and 2004 were predictable with an agreement rate of 91.67%. They also identified 20 amino acid positions whose changes significantly contribute to the development of new variants[62]. Yang et al. developed a learning sparse algorithm called AntigenCO, which uses the HA1 protein of the H3N2 virus to identify the superior determinant properties of the antigenic profiles in serological data. In addition to single mutations, their methods also used multiple simultaneous mutations or co-evolved sites in order to predict antigenicity. The prediction accuracy of the method studied in their work was 90.05%[63]. Łuksza et al. developed a model based on a mapping between the HA sequences and viral fitness. According to them, mutation at the epitope region is considered a positive fitness effect as it can induce cross-immunity between the flu strains. Conversely, the incidence of mutation outside of the epitope site is regarded as a fitness cost. Using this system, viral clades that are involved in future epidemics can be predicted[64]. Yin et al. developed a computational method to predict suitable strains for vaccination by generating time-series training samples with splittings and embeddings. Their method uses the Recurrent Neural Network (RNN), which helps in performing sequence-to-sequence prediction using H3N2 flu strains identified between 2006 and 2017. The suggested strains by their method had 98% similarity to the recommended vaccine strains[65]. In the current work, we study an anomaly detection-based approach to predicting Influenza mutations. The study results suggest some important alteration positions in the HA of influenza H1N1, H3N2, and H5N1. This multidisciplinary study shows promising and reliable results in the mutation prediction of the influenza virus in comparison with mentioned previous studies. Based on the nature of anomaly detection, this approach seems to be appropriate for mutation prediction analysis in viral genomes for further studies. By considering the complete HA gene and the ROC curves, our current model represents great performance. However, in the assessment of important antigenic sites, more optimization would be critical for further studies. Furthermore, one of the major limitations in the current study was the limited available sequences for different years. This limitation reflects a minor problem

in our current research by considering the parameter T. Meanwhile, great efforts during the time of the SARS-CoV-2 pandemic and the availability of a great amount of the primary sequence data could be promising for further studies. By considering the advancement of high throughput sequencing techniques, this limitation in sequences can be solved in the future. Another limitation of our current study was the evaluation of all lineages and clades of any H type influenza. Different clades or lineages in H1N1 have unique evolutions and need more focus in future studies in light of the improving primary sequence limitation. In addition, another limitation in the analysis of the H5N1 mutations, which needs to be mentioned, is the lack of H5N1 reports in human hosts during 2016. Furthermore, this needs to be considered that, another limitation in our current study is the theory of mutational pressure in the prediction of positions[66]. This could be a lead for future more developed models. It could be also noted that by using our model we cannot differentiate between mutation-prone gene locations and the positive pressure of natural selection in our sequence. This could be a challenging perspective for future studies and developing more accuracy in models.

Recently, this RNN model has been used to predict the antigenic changes of the Severe respiratory syndrome coronavirus 2 (SARS-COV-2) by Sawmya et al[67]. They used a pipeline method to predict mutations in some regions of the SARS-COV-2 gene. At first, genes are classified according to the country, and their phylogenetic tree is obtained. In the next step, using algorithmic methods and methods used in machine learning, positions of genes that are distinct in terms of characteristics are obtained. Finally, the occurrence of mutations in these sites is predicted by using the CNN-RNN network. Considering the SARS-CoV-2, the current study's data is just preliminary and just a perspective for further studies. This preliminary data represents low productivity in all other potential positions for mutations in the SARS-CoV-2 S gene. But highlights an important position for mutation in amino acid number 500 (represents N501Y mutation in the Alpha variant).

Methods

The influenza virus HA and SARS-CoV-2 RBD database for amino acid sequences. The influenza virus sequences are obtained from the NCBI influenza database http://www.ncbi.nlm.nih.gov/genomes/FLU/FLU.html. The amino acid sequences were obtained without the limitation in time for the report. The amino acid sequences for the human hosts in the north hemisphere referring to the H1 of the H1N1 strains were downloaded. Only the full-length sequences for the HA region were used for the Multiple Sequence Alignment (MSA). The MSA is performed by the MAFFT algorithm. For the experiments on SARS-CoV-2, we used the COVID-19 data portal provided by the European COVID-19 Data Platform and EMBL's European Bioinformatics Institute (EMBL-EBI)[68].

Proposed method. The mutation prediction problem can be formulated as the time series anomaly detection. For getting the time series data out of the raw amino-acid sequence, a preprocessing method is employed. Firstly, the ambiguous amino acids are replaced with one of the common 20 amino acids. Then, the amino acid sequences are clustered, and the primary time series sequence is obtained from them. Lastly, the amino acids are represented in some feature space using the prot2vec[69] method. In order to formulate the problem as the anomaly detection, we selected temporal attention-based RNN[70] as the backbone that is trained with a DeepSAD loss function. This lets us not only preserve the sequential aspect of the input but also benefit from the anomaly detection algorithms. Temporal attention-based RNN has shown significantly better performance compared to LSTM networks in preserving long-term dependencies. Besides, owing to the use of the attention mechanism, it only focuses on the important features of its input, which is helpful in complex problems such as mutation prediction tasks. In addition, the DeepSAD loss function helps our model not only focus on the important parts of the input sequence but also encapsulate them in a compact representation space. This is exactly what we look for in the mutation prediction problem. Our codes are made publicly available at https://github.com/rohban-lab/Tempel-HSC- for the sake of reproducibility.

Preprocessing. The preprocessing, similar to Yin et al.[41], includes three major steps: 1. replacing obscure amino acids; 2. acquiring time series sequences; 3. representing the amino acids in a feature space. Proteins consist of 20 common amino acids. Due to errors in reading the protein sequence in the stated datasets, there might be some ambiguous amino acids or letters, such as 'B', 'Z', 'J', and 'X', in the sites that are needed to be replaced with a probable amino acid. In this step, 'Z' will be randomly replaced with one of 'D' or 'N' amino acids, 'Z' with one of 'E' or 'Q', 'J' with one of 'I' or 'L', and 'X' with all amino acids. After replacement, for obtaining the time series sequences, samples in each year are divided into k clusters using the k-means algorithm. For our experiments, similar to Yin et al., we set $k = 3$. Then, by a similarity metric such as the Euclidean distance, for each cluster in a year, the closest cluster in the next year is selected. By finding the closest cluster pairs in two consecutive years, multiple sequences of clusters are formed. For example, assume that in the year t, the cluster M is closest to the cluster N in the year $t + 1$, and the cluster O in the year $t + 2$ is closest to the cluster N. So $M, N, O, \ldots$ will make a cluster sequence. From each cluster in a cluster sequence, a protein sample is randomly selected, and hence from this sequence, a time series sequence of data is created. In feature space representation, each amino acid is converted to a vector using ProtVec[69]. The representation is calculated by the following equation:

$$x_i^t = \frac{\sum_{j=-d}^{d} v_{i+j}^t}{2d+1} \tag{6}$$

where v_i^k is the vector output of ProtVec for the i-th amino acid in a sample gathered in the year t. d shows the number of amino acid neighbors. Here, like Yin et al., representations are evaluated in 3-grams, so d is set to

1. After these steps, specific site positions, which are also known as epitopes, are only selected in each protein sequence, and the RNN model is only trained and tested on these sites.

To generate the label for position i in the sequences, assuming the dataset sequences are until year $T+1$, the nucleotide value in position i at time $T+1$ is compared to the nucleotide value in position i at time T. If the values differ, it indicates a mutation and hence it is labeled as 0 (anomaly), otherwise the data is labeled 1. Afterward, for training and testing, only the sequences up to year T are given as the input to the model.

Temporal attention. Given a specific time series sequence $(x_1, x_2, \ldots, x_t)$, a temporal attention mechanism is used for encoding the sequence. Architectures such as LSTM and GRU have a performance decay when the input length increases. By applying the temporal attention mechanism to the mentioned architectures, we are able to consider each element of the sequence in the final encoding of the sequence and overcome this issue. In this mechanism, in order to calculate attention weights w_{ji}, first, an attention function f takes the hidden state h_i for the time i and the cell state s_{j-1} for the time $j-1$ from the LSTM cell and outputs a scalar value e_{ji}:

$$e_{ji} = f(s_{j-1}, h_i). \tag{7}$$

The attention function f is defined as:

$$f(s_{j-1}, h_i) = W_{attn}[s_{j-1}; h_i] + b_{attn}, \tag{8}$$

Where W_{attn} and b_{attn} are learned through the training process. Then using a softmax function, the weight w_{ti} is calculated and applied to the hidden states as below:

$$w_{ji} = \frac{\exp(e_{ji})}{\sum_{k=1}^{j-1} \exp(e_{jk})} \tag{9}$$

$$z_j = \sum_{k=1}^{j-1} w_{jk} h_k, \tag{10}$$

where z_j is the context vector for the time t. In the last step, using the following equation, the encoded vector $\hat{h}_j$ at time j is calculated:

$$\hat{h}_j = tanh(W_{\hat{h}}[z_j; h_j] + b_{\hat{h}}), \tag{11}$$

where $W_{\hat{h}}$ and $b_{\hat{h}}$ are learned through the training process. In the next step, the DeepSAD loss discussed in the next section is applied to the encoded vector $\hat{h}_T$, where T is the length of the input time series sequence.

DeepSAD loss. As mentioned earlier, we use DeepSAD loss function[29] on temporal attention RNN to capture normal features. DeepSAD loss function is defined in Eq. (12).

$$\frac{1}{m+n}\sum_{i=1}^{n} \|\phi(\hat{h}_i;\theta) - c\|^2 + \frac{\eta}{m+n}\sum_{j=n+1}^{m+n} (\|\phi(\hat{h}_j;\theta) - c\|^2)^{-1} + \frac{\lambda}{2}\sum_{l=1}^{L} \|\theta^l\|_F^2 \tag{12}$$

Here, $\phi(.;\theta)$ is a neural network, our temporal attention-based RNN and θ denotes the parameters of ϕ. Also, c is a hyper-parameter that is set before the training process. By minimizing the loss function, normal samples are centralized in a minimum volume hyper-sphere with a center c, while abnormal samples are forced to be out of it. Also, η is a hyper-parameter that can assign extra weight to the abnormal samples to mitigate the imbalanced training samples issue. Finally, n and m denote the number of normal and abnormal training samples, respectively.

As mentioned in Ruff et al.[30] use of the radial basis function in the DeepSAD loss produces better results. Therefore, the Eq. (13) that is called the hyper-sphere classifier (HSC) loss function is used in all our experiments:

$$\frac{1}{m+n}\left(\sum_{i=1}^{m+n} y_i \|\phi(\hat{h}_i;\theta)\|^2 - (1-y_i)\log\left(1 - \exp\left(-\left(\sqrt{\|\phi(\hat{h}_i;\theta)\|^2 + 1} - 1\right)\right)\right)\right) \tag{13}$$

HSC forces normal data to be mapped near the origin, that is, the center is set to zero. Since all data in our dataset has a label, the dataset is assumed to be a set of pairs (x_i, y_i). Here, $m+n$ is the number of training samples, and $y \in \{0, 1\}$, where $y = 1$ shows normal data (in our case, an unmutated virus), and $y = 0$ shows a mutated sample.

Ablation. As described above, RNNs are one of the mechanisms that can produce satisfactory results in the case of sequence transduction. However, with all their efficiencies, they have the problem of capturing long-term dependencies, which is mainly because of the fact that input sequences are passed at each step in a chain, and if the chain becomes longer, it will be more probable that the information gets lost. Unlike RNNs, Transformers can be very efficient in retaining long-term dependencies. Transformers are a kind of Neural Network architecture that has become popular due to their performance in various fields. They were introduced to perform

sequence transduction in which there are dependencies between input elements. Its architecture consists of an encoder and a decoder. The encoder has some layers that together generate encodings that summarize each element in relation to all other elements in the sequence, through the self-attention mechanism. Each layer consists of a self-attention mechanism and a Feed-Forward Neural Network.

When an input sequence is fed into the encoder, each input part flows through each of the two layers of the encoder. They first go through a self-attention mechanism, which learns three weight matrices; the query weights W_Q, the key weights W_K, and the value weights W_V. For each input element i, the input x_i is multiplied by each of the three vectors. This will produce a query vector $q_i = x_i W_Q$, a key vector $k_i = x_i W_K$, and a value vector $v_i = x_i W_V$. The attention weight a_{ij} from the input element i to the input element j can be obtained by applying the dot product between q_i and k_j. Then, the attention weights are divided by d_k, which is the dimension of the key vectors. This leads to having more stable gradients, which is an advantage over RNNs. Then, we pass the results through a softmax function in order to normalize the weights. Next, we multiply each value vector by the softmax score, and at the end, we sum up these weighted value vectors as follows:

$$Attention(q_i, k_i, v_i) = Softmax\left(\frac{q_i k_i^T}{\sqrt{d_k}}\right) v_i. \tag{14}$$

The resulting vector would then be sent along with the Feed Forward Neural Network. The decoder functions in a similar way. The decoder module contains some layers that take all the encodings and use them to generate an output sequence. Each decoder layer has both components of an encoder layer but it also has an attention layer between the two components. This helps the decoder focus on relevant parts of the input sequence.

Conclusion and future work

In this research, we have shown the effectiveness of anomaly detection approaches in predicting Influenza mutations by conducting extensive experiments. Due to the extreme imbalance between the training samples in such problems, anomaly detectors can extract useful information more efficiently and find shared, more robust features for the unmutated samples. Also, the results when the parameter T is small advocate the mentioned merits of our approach. For future works, we want to extend our experiments and provide some results on SARS-CoV-2 (severe acute respiratory syndrome coronavirus 2) mutation prediction, which can be very helpful in understanding its behavior for further applications.

Data availability

The data that support the findings of this study are available from Tempel[41]. The data is hosted at https://drive.google.com/drive/folders/1-pJGBsVfIqCEizetTQe43OQJvkmhocdW.

Received: 16 July 2022; Accepted: 5 September 2023
Published online: 11 September 2023

References

1. Bush, R. M., Bender, C. A., Subbarao, K., Cox, N. J. & Fitch, W. M. Predicting the evolution of human influenza A. *Science* **286**, 1921–1925 (1999).
2. Banning, M. Influenza: Incidence, symptoms and treatment. *Br. J. Nurs.* **14**, 1192–1197 (2005).
3. Simonsen, L. *et al.* The impact of influenza epidemics on mortality: Introducing a severity index. *Am. J. Public Health* **87**, 1944–1950 (1997).
4. Webster, R. G., Bean, W. J., Gorman, O. T., Chambers, T. M. & Kawaoka, Y. Evolution and ecology of influenza A viruses. *Microbiol. Mol. Biol. Rev.* **56**, 152–179 (1992).
5. Bedford, T. *et al.* Global circulation patterns of seasonal influenza viruses vary with antigenic drift. *Nature* **523**, 217–220 (2015).
6. Chen, J.-M. *et al.* Exploration of the emergence of the victoria lineage of influenza b virus. *Arch. Virol.* **152**, 415–422 (2007).
7. Bodewes, R. *et al.* Recurring influenza B virus infections in seals. *Emerg. Infect. Dis.* **19**, 511 (2013).
8. Krammer, F. The human antibody response to influenza A virus infection and vaccination. *Nat. Rev. Immunol.* **19**, 383–397 (2019).
9. Luoh, S.-M., McGregor, M. & Hinshaw, V. Hemagglutinin mutations related to antigenic variation in h1 swine influenza viruses. *J. Virol.* **66**, 1066–1073 (1992).
10. Caton, A. J., Brownlee, G. G., Yewdell, J. W. & Gerhard, W. The antigenic structure of the influenza virus a/pr/8/34 hemagglutinin (h1 subtype). *Cell* **31**, 417–427 (1982).
11. Brownlee, G. & Fodor, E. The predicted antigenicity of the haemagglutinin of the 1918 Spanish influenza pandemic suggests an avian origin. *Philos. Trans. R. Soc. Lond. Ser. B* **356**, 1871–1876 (2001).
12. Shen, J., Ma, J. & Wang, Q. Evolutionary trends of a (h1n1) influenza virus hemagglutinin since 1918. *PLoS ONE* **4**, e7789 (2009).
13. Buckland, B. C. The development and manufacture of influenza vaccines. *Hum. Vaccines Immunother.* **11**, 1357–1360 (2015).
14. Ampofo, W. K. *et al.* Strengthening the influenza vaccine virus selection and development process: Report of the 3rd who informal consultation for improving influenza vaccine virus selection held at who headquarters, Geneva, Switzerland, 1–3 April 2014. *Vaccine* **33**, 4368–4382 (2015).
15. Lin, Y. *et al.* Optimisation of a micro-neutralisation assay and its application in antigenic characterisation of influenza viruses. *Influenza Respir. Viruses* **9**, 331–340 (2015).
16. Tabibzadeh, A. *et al.* Evolutionary study of COVID-19, severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) as an emerging coronavirus: Phylogenetic analysis and literature review. *Vet. Med. Sci.* **7**, 559–571 (2020).
17. Kumar, S., Nyodu, R., Maurya, V. K. & Saxena, S. K. Morphology, genome organization, replication, and pathogenesis of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). In *Medical Virology: From Pathogenesis to Disease Control* 23–31 (Springer Singapore, 2020).
18. Bourgonje, A. R. *et al.* Angiotensin-converting enzyme 2 (ACE2), SARS-CoV-2 and the pathophysiology of coronavirus disease 2019 (COVID-19). *J. Pathol.* **251**, 228–248 (2020).
19. Xiaojie, S., Yu, L., Lei, Y., Guang, Y. & Min, Q. Neutralizing antibodies targeting SARS-CoV-2 spike protein. *Stem Cell Res.* **50**, 102125 (2020).

20. Tsipras, D., Santurkar, S., Engstrom, L., Ilyas, A. & Madry, A. From imagenet to image classification: Contextualizing progress on benchmarks. In *International Conference on Machine Learning* 9625–9635 (PMLR, 2020).
21. Kowsari, K. *et al.* Text classification algorithms: A survey. *Information* **10**, 150 (2019).
22. Zhang, H.-B. *et al.* A comprehensive survey of vision-based human action recognition methods. *Sensors* **19**, 1005 (2019).
23. Chalapathy, R. & Chawla, S. Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407 (2019).
24. Ruff, L. *et al.* Deep one-class classification. In *International Conference on Machine Learning* 4393–4402 (PMLR, 2018).
25. Salehi, M. *et al.* Arae: Adversarially robust training of autoencoders improves novelty detection. arXiv preprint arXiv:2003.05669 (2020).
26. Akcay, S., Atapour-Abarghouei, A. & Breckon, T. P. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian Conference on Computer Vision* 622–637 (Springer, 2018).
27. Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U. & Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging* 146–157 (Springer, 2017).
28. Goyal, S., Raghunathan, A., Jain, M., Simhadri, H. V. & Jain, P. Drocc: Deep robust one-class classification. In *International Conference on Machine Learning* 3711–3721 (PMLR, 2020).
29. Ruff, L. *et al.* Deep semi-supervised anomaly detection. arXiv preprint arXiv:1906.02694 (2019).
30. Ruff, L., Vandermeulen, R. A., Franks, B. J., Müller, K.-R. & Kloft, M. Rethinking assumptions in deep anomaly detection. arXiv preprint arXiv:2006.00339 (2020).
31. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
32. Schuster, M. & Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45**, 2673–2681 (1997).
33. Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014).
34. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R. & Schmidhuber, J. Lstm: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **28**, 2222–2232 (2016).
35. Salehi, M. *et al.* A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. arXiv preprint arXiv:2110.14051 (2021).
36. Chong, P., Ruff, L., Kloft, M. & Binder, A. Simple and effective prevention of mode collapse in deep one-class classification. In *2020 International Joint Conference on Neural Networks (IJCNN)* 1–9 (IEEE, 2020).
37. Sabokrou, M., Khalooei, M., Fathy, M. & Adeli, E. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 3379–3388 (2018).
38. Zaheer, M. Z., Lee, J.-h., Astrid, M. & Lee, S.-I. Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 14183–14193 (2020).
39. Perera, P., Nallapati, R. & Xiang, B. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2898–2906 (2019).
40. Salehi, M., Eftekhar, A., Sadjadi, N., Rohban, M. H. & Rabiee, H. R. Puzzle-ae: Novelty detection in images through solving puzzles. arXiv preprint arXiv:2008.12959 (2020).
41. Yin, R., Luusua, E., Dabrowski, J., Zhang, Y. & Kwoh, C.-K. Tempel: Time-series mutation prediction of influenza a viruses via attention-based recurrent neural networks. *Bioinformatics*https://doi.org/10.1093/bioinformatics/btaa050 *(2020).*
42. Centers for disease control and prevention (2016).
43. Qiagen clc genomics workbench 20.0 (2000).
44. de la Rosa-Zamboni, D. *et al.* Molecular characterization of the predominant influenza a (h1n1) pdm09 virus in Mexico, December 2011–February 2012. *PLoS ONE* **7**, e50116 (2012).
45. Zost, S. J. *et al.* Identification of antibodies targeting the h3n2 hemagglutinin receptor binding site following vaccination of humans. *Cell Rep.* **29**, 4460–4470 (2019).
46. Golan, I. & El-Yaniv, R. Deep anomaly detection using geometric transformations. arXiv preprint arXiv:1805.10917 (2018).
47. Bergman, L. & Hoshen, Y. Classification-based anomaly detection for general data. arXiv preprint arXiv:2005.02359 (2020).
48. Huang, C., Ye, F., Zhang, Y., Wang, Y.-F. & Tian, Q. Esad: End-to-end deep semi-supervised anomaly detection. arXiv preprint arXiv:2012.04905 (2020).
49. Wu, G. & Yan, S.-M. Mutation trend of hemagglutinin of influenza A virus: A review from a computational mutation viewpoint. *Acta Pharmacol. Sin.* **27**, 513–526 (2006).
50. Shirvani, E., Paldurai, A., Varghese, B. P. & Samal, S. K. Contributions of ha1 and ha2 subunits of highly pathogenic avian influenza virus in induction of neutralizing antibodies and protection in chickens. *Front. Microbiol.* **11**, 1085 (2020).
51. Shen, X., Zhang, X. & Liu, S. Novel hemagglutinin-based influenza virus inhibitors. *J. Thorac. Dis.* **5**, S149 (2013).
52. Knipe, D. *et al. Fields Virology* (Lippincott Williams & Wilkins, 2013).
53. Krammer, F. & Palese, P. Influenza virus hemagglutinin stalk-based antibodies and vaccines. *Curr. Opin. Virol.* **3**, 521–530 (2013).
54. Ning, T. *et al.* Antigenic drift of influenza A (h7n9) virus hemagglutinin. *J. Infect. Dis.* **219**, 19–25 (2019).
55. De, A. Molecular evolution of hemagglutinin gene of influenza A virus. *Front. Biosci.* **10**, 101–118 (2018).
56. Webster, R., Laver, W., Air, G. & Schild, G. Molecular mechanisms of variation in influenza viruses. *Nature* **296**, 115–121 (1982).
57. Wu, G. & Yan, S. Timing of mutation in hemagglutinins from influenza A virus by means of unpredictable portion of amino-acid pair and fast Fourier transform. *Biochem. Biophys. Res. Commun.* **333**, 70–78 (2005).
58. Doud, M. B., Lee, J. M. & Bloom, J. D. How single mutations affect viral escape from broad and narrow antibodies to h1 influenza hemagglutinin. *Nat. Commun.* **9**, 1–12 (2018).
59. Lyons, D. M. & Lauring, A. S. Mutation and epistasis in influenza virus evolution. *Viruses* **10**, 407 (2018).
60. Thyagarajan, B. & Bloom, J. D. The inherent mutational tolerance and antigenic evolvability of influenza hemagglutinin. *elife* **3**, e03300 (2014).
61. Morris, D. H. *et al.* Predictive modeling of influenza shows the promise of applied evolutionary biology. *Trends Microbiol.* **26**, 102–118 (2018).
62. Liao, Y.-C., Lee, M.-S., Ko, C.-Y. & Hsiung, C. A. Bioinformatics models for predicting antigenic variants of influenza A/h3n2 virus. *Bioinformatics* **24**, 505–512 (2008).
63. Yang, J., Zhang, T. & Wan, X.-F. Sequence-based antigenic change prediction by a sparse learning method incorporating co-evolutionary information. *PLoS ONE* **9**, e106660 (2014).
64. Łuksza, M. & Lässig, M. A predictive fitness model for influenza. *Nature* **507**, 57–61 (2014).
65. Yin, R., Zhang, Y., Zhou, X. & Kwoh, C. K. Time series computational prediction of vaccines for influenza a h3n2 with recurrent neural networks. *J. Bioinform. Comput. Biol.* **18**, 2040002 (2020).
66. Sueoka, N. Directional mutation pressure and neutral molecular evolution. *Proc. Natl. Acad. Sci. USA* **85**, 2653–2657 (1988).
67. Sawmya, S. *et al.* Analyzing hcov genome sequences: Predicting virulence and mutation. *bioRxiv* 2020–06 (2021).
68. The European Covid-19 Data Platform. https://www.covid19dataportal.org/the-european-covid-19-data-platform.
69. Asgari, E. & Mofrad, M. R. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS ONE* **10**, e0141287 (2015).
70. Qin, Y. *et al.* A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint arXiv:1704.02971 (2017).

Author contributions

A.G., A.M.C, M.S., A.T., and M.H.Rohban. designed the conceptual solution. A.G. and A.M.C. implemented and tested the idea. A.T., P.Y., M.H.Razizadeh, Moein E., and Maryam E. interpreted and discussed the results. A.G., M.S., and A.T. wrote the main manuscript. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-023-42089-y.

Correspondence and requests for materials should be addressed to M.H.R.

Reprints and permissions information is available at www.nature.com/reprints.

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