UDC 004.896
DOI: 10.20535/SRIT.2308-8893.2023.3.03

RESEARCH ON HYBRID TRANSFORMER-BASED AUTOENCODERS FOR USER BIOMETRIC VERIFICATION

M.P. HAVRYLOVYCH, V.Y. DANYLOV

Abstract. Our current study extends previous work on motion-based biometric verification using sensory data by exploring new architectures and more complex input from various sensors. Biometric verification offers advantages like uniqueness and protection against fraud. The state-of-the-art transformer architecture in AI is known for its attention block and applications in various fields, including NLP and CV. We investigated its potential value for applications involving sensory data. The research proposes a hybrid architecture, integrating transformer attention blocks with different autoencoders, to evaluate its efficacy for biometric verification and user authentication. Various configurations were compared, including LSTM autoencoder, transformer autoencoder, LSTM VAE, and transformer VAE. Results showed that combining transformer blocks with an undercomplete deterministic autoencoder yields the best performance, but model performance is significantly influenced by data preprocessing and configuration parameters. The application of transformers for biometric verification and sensory data appears promising, performing on par with or surpassing LSTM-based models but with lower inference and training time.

Keywords: biometric verification, transformers, variational autoencoder, transformer autoencoder.

INTRODUCTION

The usage of various deep learning algorithms boosted and enabled various AI and machine learning fields and applications. The biometric field was no exception, specifically with the growth and significant adoption of various electronic devices such as smartphones, bracelets, watches, etc. One of the important areas where biometric data is utilised is security, verification and authentication. Much research was conducted in this field to discover and provide deep learning architectures that will be able to build efficient and reliable systems feasible for usage in real life.

Traditional methods, such as passwords and PINs, are prone to breaches and hacking, as well as are challenging to manage, which lead us to the exploration of more secure and user-friendly alternatives. However, the effectiveness of biometric verification is contingent on the ability to process and interpret complex biometric data accurately. Deep learning approached, which can generalize over large data samples and be high-performant, is a solution to solve the problem. Specifically, combining autoencoders and transformer attention layers, a novel deep learning approach, has shown promise in enhancing the performance of biometric verification systems. However, this approach is still not widely presented in biometric verification and continuous authentication research.

The relevance of this research lies in developing more secure and efficient user authentication methods. By enhancing the performance of biometric verification systems, we can provide a more secure and convenient alternative to traditional authentication methods.

The object of this research is the application of autoencoders combined with transformer attention layers in biometric verification and continuous authentication.

This study investigates the effectiveness of autoencoders combined with transformer attention layers for biometric verification and continuous authentication. We aim to assess whether this novel approach can improve the performance and efficiency of biometric verification systems, thereby contributing to the development of more secure and user-friendly authentication methods.

LITERATURE REVIEW

In [1], the authors convey an in-depth survey on which deep learning and machine learning models are used for biometric verification. There is extensive research on hybrid models, such as extracting features with the CNN model and conducting authentication with some machine learning models, such as SVM or One-Class SVM or LSTM block with further Stochastic Gradient Descent (SGD) classifier. Another quite popular solution is using LSTM model architecture, which is self-explainable as biometric in many cases is sensory data with a sequential structure. Specifically for the motion or gait patterns, the hybrid architecture LSTM + CNN is popular, which outperforms the LSTM or CNN separately [1; 2]. Overall it is noticed that hybrid architectures provide a boost in performance and are widely adopted in biometric authentication. It is worth noting that there is no clear distinction between supervised and unsupervised approaches in the paper, and all of them are compared altogether, which is essential for the context of the constraints and limitations of the implemented verification system. Our interest is in unsupervised approaches as they provide a solution in real cases when there is no access to other users' data (as it will be due to data privacy), contrary to supervised models.

The data nature causes the popularity of LSTM applications for sensory data, but not only recurrent architectures can handle sequences. The transformer architecture [3] was initially adopted in natural language processing (NLP) tasks and almost replaced the recurrent neural networks in that field [4].

Transformers' way of consuming sequences provided faster training and inference and better generalization capabilities for sequences as it does not have an issue of forgetting input in case of long input, as the sequence was consumed as a whole instantly and not chunk by chunk. On the other hand, the architecture requires fixed sequence length and sequences with lengths higher than the model support will not be processed. As the transformers were great with sequence data — they slowly started being used in other fields, such as CV and time series. In the [5; 6], authors review the effectiveness of transformers for time series data and compare various transformer types, which show pretty decent results.

Nevertheless, RNNs are still holding their place in the time series field, as they are better at capturing the autoregressive nature of time series signals. Both models have pros and cons, and at the end of the day, each can bring something to the table. In [7], authors show that LSTM with attention layer outperforms the transformer-based model for time series tasks, which supports the idea that hybrid models create more performant and robust deep learning systems.

In biometric fields, transformers were used for human activity recognition (HAR) problems [8; 9]. The authors proposed a HAR transformer, which solves the time series classification problem.

The choice of the approach and model architecture for biometric verification depends on which type of authentication system we want to build. Authentication can be implicit and explicit, as well as continuous or more discrete. We are interested in implicit continuous authentication, generally the unsupervised approach. The overall model architecture used for such tasks is autoencoder. We have reviewed and experimented with the usage of autoencoders for biometric verification tasks in our previous research [10]. In another paper, we reviewed which sensor data signal contributes the most to creating a distinctive user pattern [11].

In [12], the VAE-based system was proposed to solve the text keystroke authentication when the training is done on the English typing data and evaluating the Korean typing data from the same users. This may show that the model learns the pattern of the user uniqueness and not the different patterns related to activities. A deep LSTM-based autoencoder is proposed in [13] for anomaly detection in ECG signals. In contrast, in [14], adversarial autoencoder [15], which is the combination of autoencoder with generative adversarial networks (GAN), was used for the health monitoring of ECG and for detecting abnormal data points, which by the authors outperformed LSTM and VAE architectures. The autoencoder with attention mechanism, placed between encoder and decoder blocks to learn relations on the latent space feature representations, is proposed in [16] for ECG data anomaly detection.

However, the LSTM-based architecture still was more performant and better at capturing time series data. In [17], the authors proposed the attentive adversarial autoencoder for user authentication. Compared to approaches like one-class SVM, LSTM and HMM, the autoencoder-based solution achieved the highest performance in terms of qualitative metrics and time performance. In [18], the purely transformer-based architecture is used for detecting anomalies in ECG series, which is also shown to be a viable option.

Autoencoder and its various modification of it are widely used and researched in the area of intelligent fault diagnosis and prognosis for industrial systems [19]. In this area, autoencoders help to prevent system failure processing, like wind turbine equipment or other complex systems, processing the multiple modality data, such as acoustic and vibration signals [20]. Stacked autoencoder architecture is quite famous for fault diagnosis, where multiple encoders and decoders are stacked on top of each other, which may help the neural network to recognize data trends and patterns better.

We want further review and experiment with various autoencoder-based architecture sand specifically review the possibility of incorporating elements from other architecture to see whether it will impact the performance. As the transformer-based architecture is still state-of-the-art in many fields, though it was proposed some time ago, and multiple other research incorporate it for various biometric-related tasks, such as health monitoring – we would like to experiment with how it will impact metrics in biometric verification tasks, and whether it will reduce the inference time, as a transformer, due to the way how they process sequence should be faster than RNN.

MATERIALS AND METHODS

As a baseline model with which we will compare other experiments, an LSTM autoencoder will be used. The autoencoder is an artificial neural network for learning hidden internal representations and features of input data. It consists of two main parts: an encoder that compresses the input into a latent-space representation and a decoder which reconstructs the input from the latent space. During training autoencoder learns to minimize the difference between the input and the reconstructed output. The optimization task objective is to minimize this difference, called the reconstruction error:

$$E = \sqrt{\sum_{i=1}^{n} \left\| x_i - d_{\varphi}(e_{\theta}(x_i)) \right\|},$$

where $x_1, \ldots, x_n$ is data rows, and the functions d_{φ} and e_{θ} represent the encoder and decoder, respectively, with some parameters φ and θ.

Autoencoder can be considered as a high-level neural network architecture, as it does not limit what architectural elements should or should not be in the encoder and decoder. However, there are some types of autoencoders that specify some limitations on the architecture of the autoencoder or some of its configurations. For example, a sparse autoencoder should have a dimension of latent space higher than the input dimension; the denoising autoencoder puts the requirement for adding noise to the input data; the contractive autoencoder specifies the optimization loss.

Variational Autoencoder (VAE) is somewhat different from other autoencoder types, as it maps the input data not to the fixed latent space representation, but the Gaussian distribution with some parameters (mean and variance). Thus, it allows us to present our input data points in probabilistic manner. This model architecture is close to the generative AI algorithms we reconstruct our data sampling it from out latent distribution, so in fact generating it [21].

The encoder part of the VAE is defined as:

1. Encoder:

$$\mu = W_{\mu} * h + b_{\mu};$$

$$\log(\sigma^2) = W_{\sigma} * h + b_{\sigma}.$$

2. Reparameterization Trick:

$$z = \mu + \sigma \odot \varepsilon, \text{ where } \varepsilon \sim N(0, I).$$

3. Decoder:

$$p(x \mid z) = f_{\text{dec}}(z).$$

4. Loss Function:

$$L = E[\log p(x / z)] - D_{KL}(Q(z / x) \| P(z)),$$

where h is the output of the encoder's hidden layer; W_{μ}, W_{σ}, b_{μ}, and b_{σ} are the weights and biases for the mean and log-variance, respectively; μ and σ are the mean and standard deviation of the latent variable z; ε is a random variable sampled from a standard normal distribution; $\odot$ denotes element-wise multiplication;

f_{dec} is the decoder function; $p(x/z)$ is the probability of the data given the latent variable; $Q(z/x)$ is the approximate posterior distribution; $P(z)$ is the prior distribution (standard normal distribution in the case of VAEs); $D_{KL}(...)$ is the Kullback–Leibler divergence, which measures the difference between two probability distributions; $E[\ldots]$ denotes the expectation; L is the loss function that the VAE aims to minimize.

These formulas represent the core of the VAE. The encoder generates the parameters of the latent variable's distribution, the reparameterization trick is used to sample from this distribution, and the decoder generates the data from the sampled latent variable. The loss function consists of the reconstruction loss (the first term) and the regularization term (the second term).

Neural network building blocks. As autoencoder is a high-level architecture – it may be constructed from any neural network units which are suitable for the given problem and data input.

Long Short-Term Memory (LSTM). LSTM is a type of recurrent neural network (RNN) that can learn and remember over long sequences and is not that by the vanishing gradient problem, as just RNN. It achieves this by using a series of "gates". These blocks collectively decide what information should be kept or discarded.

The LSTM cell can be defined by the following set of equations:

Forget gate:

$$f_t = \sigma(W_f * (h_{t-1}, x_t) + b_f) .$$

Input gate:

$$i_t = \sigma(W_i * (h_{t-1}, x_t) + b_i) .$$

Cell update:

$$\widehat{C}_t = \tanh(W_c * (h_{t-1}, x_t) + b_c) .$$

New cell:

$$C_t = f_t * C_{t-1} + i_t * \widehat{C}_t .$$

Output gate:

$$o_t = \sigma(W_o * (h_{t-1}, x_t) + b_o) .$$

New hidden state:

$$h_t = o_t * \tanh(C_t) .$$

Where σ is the sigmoid function, (h_{t-1}, x_t) denotes the concatenation of the input vector x_t and the previous hidden state h_{t-1}, and W and b are the weight matrices and bias vectors.

Transformers (attention unit). Transformers are a type of model that uses self-attention mechanisms and are particularly effective for tasks involving sequential data. Unlike RNNs, transformers do not require that the sequence data be processed in order, thus allowing for parallel processing of the data.

The self-attention mechanism in transformers can be defined as:

$$Q = W_q * X, K = W_k * X, V = W_v * X .$$

$$\text{Attention}(Q,K,V) = \text{soft}\max\left(\frac{Q*K^T}{\sqrt{d_k}}\right)*V\,.$$

Where Q, K, and V are the query, key, and value vectors, and d_k is the dimension of the key vector. The softmax function ensures that the weights of the different positions sum to 1.

EXPERIMENTS

Dataset. Open-source dataset [22–24], a large-scale user study with 100 volunteers to collect a wide spectrum of signals about smartphone user behaviors, including touch, gesture, and pausality of the user, as well as movement and orientation of the phone. Data from three usage scenarios on smartphones were recorded: 1) document reading; 2) text production; 3) navigation on a map to locate a destination.

The dataset contains multiple modalities input from various sensors. For our experimentation, we selected the accelerometer, gyroscope and magnetometer inputs in the dataset.

The dataset contains multiple activities, such as read and walking, read and sitting, write and walking, write and sitting, navigate the map and walking and navigate the map and sitting – overall 6 activity types. We have trained our models on some selected activity type, as well as on activity pair, like reading, navigating the map or writing and activity triplet, like sitting or walking.

For deep learning models, we split data in overlapping on 50 percent windows with a sampling of 100Hz and a length of 1s.

The original dataset is split into a 20% share for the test set and the rest for the train.

We preprocessed data in 2 ways: standart dcaling and min-max normalizing.

Sensors description. An accelerometer measures changes in velocity along one axis. The values reported by the accelerometers are measured in increments of the gravitational acceleration, with the value 1.0 representing an acceleration of 9.8 meters per second in the given direction. Depending on the direction of the acceleration, the sensor values may be positive or negative. A gyroscope measures the rate at which a device rotates around a spatial axis and is used to detect or measure direction. The magnetometer measures the strength of the magnetic field surrounding the device, allowing us to detect the device's orientation correctly [25; 26].

Metrics. The threshold formula was used as in [10]:

$$T = \sum_{i=1}^{N} MAE_i \Big/ N + std(MAE_i)\,,$$

where MAE is the mean absolute error between ground truth and predicted sample; std – standard deviation; and N is the number of samples in the training dataset.

As model evaluation metrics [27], the EER (equal error rate); FAR (false accept rate) and FRR (false reject rate) were chosen, which are typical for assessing the biometric system quality:

$$\mathrm{FAR} = \mathrm{FPR} = \mathrm{FP}/\mathrm{FP} + \mathrm{TN}\ ;$$

$$\mathrm{FRR} = \mathrm{FNR} = \mathrm{FN}/\mathrm{TP} + \mathrm{FN}\ .$$

Equal error rate is obtained by adjusting the system's detection threshold to equalize FAR and FRR. The EER is calculated using the following formula:

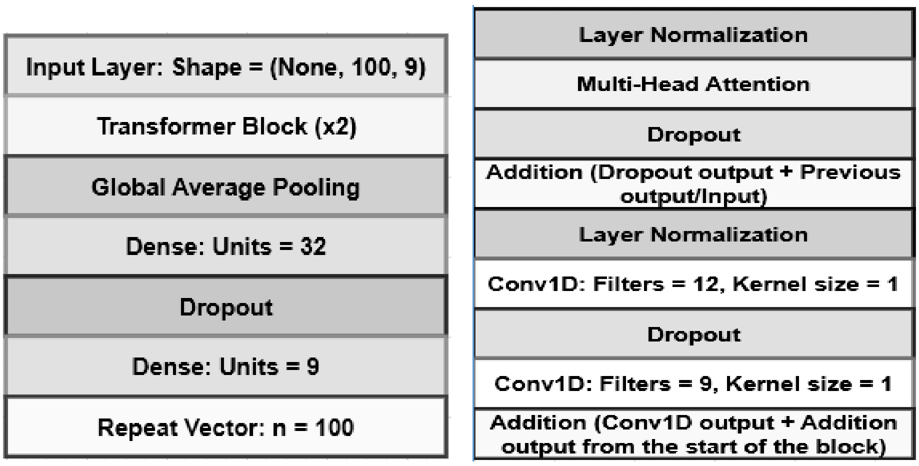
$$\mathrm{EER} = \mathrm{FAR} + \mathrm{FRR}/2\ ,$$

where $|\,\mathrm{FAR} + \mathrm{FRR}\,|$ is the smallest value [27].

The models were coded and trained in Python using Keras library with Tensorflow backend.

All models were trained in 20 epochs with Adam optimiser on the GeForce RTX 2070 GPU.

The architecture of transformer-based hybrid autoencoder used for experiments illustrated in Fig. 1.



a *b*

Fig. 1. Architecture of transformer-based hybrid autoencoder: *a* — the high-level autoencoder architecture with transformer encoder; *b* — the internal structure of transformer-based encoder with attention units

The LSTM autoencoder architecture with which the transformer-based autoencoder was compared is illustrated on Fig. 2.

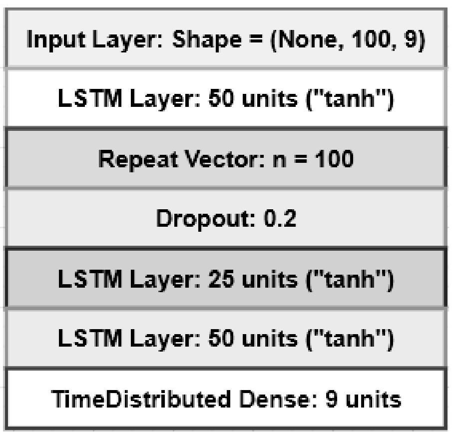


Fig. 2. LSTM autoencoder architecture

RESULTS

The experimentation results can be reviewed in the tables below.

The results for the single activity with standart scaling data preprocessing and variational-based autoencoders can be reviewed in Table 1. In Table 2 we can view the model performance for activity pairs, and results for acitivy triplet in Table 3. For the deterministic models the data was processed with min-max normalization.

As well we can view the performance time for training and inference for models in Table 4. Overall in each exeperiment for each chosen activity set data 100 model were trained.

Table 1. Average EER, FAR, FRR for 100 users for single activity

Model architecture	Average EER	Average FAR	Average FRR
Single activity — write and sitting			
LSTM VAE	5.10%	14.25%	3.28%
Transformer-VAE	4.20%	12.95%	1.72%

Table 2. Average EER, FAR, FRR for 100 users for activity pairs

Model architecture	Average EER	Average FAR	Average FRR
Activity Pair — write and walking, write and sitting			
LSTM AE	5.21%	13.30%	3.37%
Transformer AE	4.22%	13.93%	1.76%
Activity Pair – map and walking, map and sitting			
LSTM AE	6.74%	14.39%	5.00%
Transformer AE	5.87%	13.24%	3.42%

Table 3. Average EER, FAR, FRR for 100 users for activity triplet

Model architecture	Average EER	Average FAR	Average FRR
Activity Triplet — read and sitting, write and sitting, map and sitting			
LSTM AE	1.26%	12.38%	0.06%
Transformer AE	1.61%	10.97%	0.14%
Activity Triplet – read and walking, write and walking, map and walking			
LSTM AE	9.10%	12.66%	9.51%
Transformer AE	6.47%	12.37%	4.81%

Table 4. Average training and inference time for 100 users for sitting activity triplet

Model architecture	Training Time (*s*)	Inference Time (*s*)
LSTM AE (MSE loss)	90.67	43.32
Transformer-AE (MSE loss)	82.22	29.61
Difference	9.32%	31.65%

DISCUSSION

The obtained experiments results showed us that transformer architecture, specifically its central architectural unit as attention, provides performance improvement for the biometric verification task in the case of deterministic model version or generative (variational). The transformer-based autoencoder outperformed the LSTM based one in the case of training on single activity and activity pairs, which confirmed that the model performance is stable over different data inputs.

Though on sitting activity triplet, the LSTM AE slightly outperformed the Transformer AE in terms of EER and FRR but had a higher FAR rate. It shows us that the LSTM can generalize better with a larger train data sample. However, as well showing us that transformer-based autoencoders can generalize on smaller amounts of data.

It is worth noting that the transformer is significantly faster than the LSTM based model in terms of training and inference time; therefore, it is a much better fit for the edge devices like smartphones or smartwatches, where such models will be applied.

During the experimentation, we were also trying different losses and data preprocessing approaches and figured out that models are susceptible to the scale of the data input. The insightful observation was that deterministic models are great for generalization in the case of data normalization with min-max. However, in the case of standard scaling, the variational version generalizes better, which can happen due to multiple factors. First, min-max transformation can distort the data distribution in case of significant outliers in data; therefore, variational autoencoder that samples from Gaussian distribution with mean and variance will not be able to learn on the data that do not follow Gaussian distribution. On the other hand, the reason why deterministic models could not generalize well on standard scaled data was due to using as input multiple sensor signals, which may have different ranges and make it harder for neural networks that are sensitive to the range caused by the tanh activation function. Though this observation should be rigorously tested, it provides insights into how the data should be preprocessed for different architectures and how strongly the data format is coupled with the neural network.

CONCLUSIONS

We have conducted various experiments in this research and proposed and analysed the hybrid transformer-based autoencoder model architecture. The model was high-performant compared to the LSTM-based architecture and robust with different data inputs regarding amount and activity types.

Overall more than 800 neural networks were trained during the experimentation.

We have noticed that although the model architecture plays a significant part in the final metrics, the data pre-processing step is critical, and we cannot expect from deep learning model to generalise without preliminary steps. Depending on model internals, we should keep an eye on the validity of data distribution and the presence of noise and outliers in the dataset. Model type and data may also impact the selection of optimised losses, such as the used in our models' mean squared error or mean absolute error, which is more robust to the outliers, or the combination of both losses like Huber loss. During experimentation, we noticed that optimised loss may significantly add to the model's generalisation ability. However, this observation should be researched further to understand how model architecture connects with the different loss functions.

As further steps – we may consider creating the ensemble of the models in order to achieve the highest possible metric value. We can see that treating a neural network as a weak learner is possible. Though, it has a considerable amount of parameters – the discussion in the machine learning community makes us believe that it should be the auspicious direction in further neural network architecture development.

REFERENCES

1. S. Kokal, M. Vanamala, and R. Dave, "Deep Learning and Machine Learning, Better Together Than Apart: A Review on Biometrics Mobile Authentication," *Journal of Cybersecurity and Privacy*, vol. 3, no. 2, pp. 227–258, Jun. 2023. doi: 10.3390/jcp3020013.
2. Q. Cao, F. Xu, and H. Li, "User Authentication by Gait Data from Smartphone Sensors Using Hybrid Deep Learning Network," *Mathematics*, vol. 10, no. 13, p. 2283, Jun. 2022. doi: 10.3390/math10132283.
3. A. Vaswani et al., "Attention is All you Need," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017.
4. S.M. Lakew, M. Cettolo, and M. Federico, "A Comparison of Transformer and Recurrent Neural Networks on Multilingual Neural Machine Translation," in *International Conference on Computational Linguistics (COLING), 2018*.
5. A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?," in *Proceedings of the AAAI Conference on Artificial Intelligence, 2023*.
6. Q. Wen et al., "Transformers in time series: A survey," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.
7. A. Katrompas, T. Ntakouris, and V. Metsis, "Recurrence and Self-attention vs the Transformer for Time-Series Classification: A Comparative Study," in Michalowski M., Abidi S.S.R., Abidi S. (eds) *Artificial Intelligence in Medicine (AIME)*, vol. 13263. Springer, Cham, 2022. doi: 10.1007/978-3-031-09342-5_10.
8. I. Dirgová Luptáková, M. Kubovčík, and J. Pospíchal, "Wearable Sensor-Based Human Activity Recognition with Transformer Model," *Sensors*, vol. 22, no. 5, p. 1911, Mar. 2022. doi: 10.3390/s22051911.
9. A. Raza et al., "Lightweight Transformer in Federated Setting for Human Activity Recognition," *ArXiv, arXiv:2110.00244*, Nov. 4, 2022. Accessed on: July 6, 2023. [Online]. Available: https://arxiv.org/abs/2110.00244
10. M. Havrylovych and V. Danylov, "Research of Autoencoder-Based User Biometric Verification with Motion Patterns," *System Research and Information Technologies*, no. 2, 2022, pp. 128–136. doi: 10.20535/srit.2308-8893.2022.2.10.
11. M. Havrylovych, V. Danylov, and A. Gozhyj, "Comparative Analysis of using Recurrent Autoencoders for User Biometric Verification with Wearable Accelerometer," *Proceedings of the 9th International Conference "Information Control Systems & Technologies" (ICST)*, pp. 358–370, Sept. 2020.
12. F. Trad, A. Hussein and A. Chehab, "Free Text Keystroke Dynamics-based Authentication with Continuous Learning: A Case Study," in *2022 IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/*

SmartCNS), Chongqing, China, 2022, pp. 125–131. doi: 10.1109/IUCC-CIT-DSCI-SmartCNS57392.2022.00031S.

13. M. Roy, S. Majumder, A. Halder, and U. Biswas, "ECG-NET: A deep LSTM autoencoder for detecting anomalous ECG," *Engineering Applications of Artificial Intelligence*, vol. 124, p. 106484, 2023. doi: 10.1016/j.engappai.2023.106484.
14. L. Shan et al., "Abnormal ECG detection based on an adversarial autoencoder," *Frontiers in Physiology*, vol. 13, 2022. doi: 10.3389/fphys.2022.961724.
15. A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial Autoencoders," *arXiv, arXiv:1511.05644*, 2015. [Online]. Available: https://arxiv.org/abs/1511.05644.
16. A. Oluwasanmi, M.U. Aftab, E. Baagyere, Z. Qin, M. Ahmad, and M. Mazzara, "Attention Autoencoder for Generative Latent Representational Learning in Anomaly Detection," *Sensors*, vol. 22, no. 1, p. 123, Dec. 2021. doi: 10.3390/s22010123.
17. M. Hu, K. Zhang, R. You, and B. Tu, "Relative Attention-based One-Class Adversarial Autoencoder for Continuous Authentication of Smartphone Users," *arXiv, arXiv:2210.16819*, 2022. [Online]. Available: https://arxiv.org/abs/2210.16819.
18. A. Alamr and A. Artoli, "Unsupervised Transformer-Based Anomaly Detection in ECG Signals," *Algorithms*, vol. 16, no. 3, p. 152, Mar. 2023. doi: 10.3390/a16030152.
19. M. Ma, C. Sun and X. Chen, "Deep Coupling Autoencoder for Fault Diagnosis With Multimodal Sensory Data," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 1137–1145, March 2018. doi: 10.1109/TII.2018.2793246.
20. S. Qiu et al., "Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review," *Sensors*, vol. 23, no. 3, p. 1305, Jan. 2023. doi: 10.3390/s23031305.
21. L. Weng, "From Autoencoder to Beta-VAE," *Lil'Log*, Aug. 12, 2018. Accessed on: July 15, 2023. [Online]. Available: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html
22. Qing Yang et al., "A Multimodal Data Set for Evaluating Continuous Authentication Performance in Smartphones," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys '14). ACM, New York, NY, USA*, pp. 358–359. doi: 10.1145/2668332.2668366.
23. Zdeňka Sitová et al., "HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users," in *IEEE Transactions on Information Forensics and Security*, vol.11, no.5, pp. 877–892, May 2016. doi: 10.1109/TIFS.2015.2506542.
24. "A Multimodal Data Set for Evaluating Continuous Authentication Performance," *H-MOG Dataset*. Accessed on: July 15, 2023. [Online]. Available: https://hmog-dataset.github.io/hmog/
25. A. Allan, *Basic Sensors in iOS: Programming the Accelerometer, Gyroscope, and More*; 1st ed., O'Reilly Media, Beijing, Aug. 30, 2011. Accessed on: July 8, 2023. [Online]. Available: https://www.amazon.com/Basic-Sensors-iOS-Programming-Accelerometer/dp/1449308465
26. S. Balli, E.A. Sağbaş, and M. Peker, "A Mobile Solution Based on Soft Computing for Fall Detection," in *Mobile Solutions and Their Usefulness in Everyday Life*, S. Paiva, Ed. Cham: Springer International Publishing, 2019, pp. 275–294. doi: 10.1007/978-3-319-93491-4_14.
27. M. El-Abed, C. Charrier, and C. Rosenberger, "Evaluation of Biometric Systems," *New Trends and Developments in Biometrics*. InTech, Nov. 28, 2012. doi: 10.5772/52084.

Received 14.07.2023

INFORMATION ON THE ARTICLE

Mariia P. Havrylovych, ORCID: 0000-0002-9797-2863, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: mariia.havrylovych@gmail.com

Valeriy Ya. Danylov, ORCID: 0000-0003-3389-3661, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: danilov1950@ukr.net

ДОСЛІДЖЕННЯ ГІБРИДНИХ АВТОКОДУВАЛЬНИКІВ З ВИКОРИСТАННЯМ ТРАНСФОРМЕРІВ ДЛЯ БІОМЕТРИЧНОЇ ВЕРИФІКАЦІЇ КОРИСТУВАЧА / М.П. Гаврилович, В.Я. Данилов

Анотація. У дослідженні розширено попередню працю з біометричної верифікації на основі руху з використанням сенсорних даних шляхом дослідження нових архітектур та більш складних даних від різних датчиків. Біометрична верифікація дає такі переваги, як унікальність для кожного користувача і захист від шахрайства. Архітектура трансформера, одна з найсучасніших у сфері штучного інтелекту, відома своїм юнітом уваги та застосуванням у різних сферах, включаючи NLP та CV. У праці досліджено її потенційну цінність для додатків, які обробляють сенсорні дані. Дослідження пропонує гібридну архітектуру, що об'єднує блоки уваги від трансформера з різними автокодувальниками, щоб оцінити її ефективність для біометричної верифікації та аутентифікації користувача. Порівняно різні конфігурації, включно з автокодувальником LSTM, автокодувальником на базі трансформера, LSTM VAE і VAE на основі трансформера. Результати показали, що поєднання блоків трансформера із неповним детермінованим автокодувальником дає найкращі метрики, але на показники моделі також значно впливають попереднє оброблення даних і параметри конфігурації алгоритму. Застосування трансформерів для біометричної верифікації та сенсорних даних виглядає багатообіцяльним, за метриками нарівні з моделями на основі LSTM або перевершуючи їх, проте з меншими часом обробленням сигналу і навчання моделі.

Ключові слова: біометрична верифікація, транформери, варіаційний автокодувальник, автокодувальник на основі трансфомера.