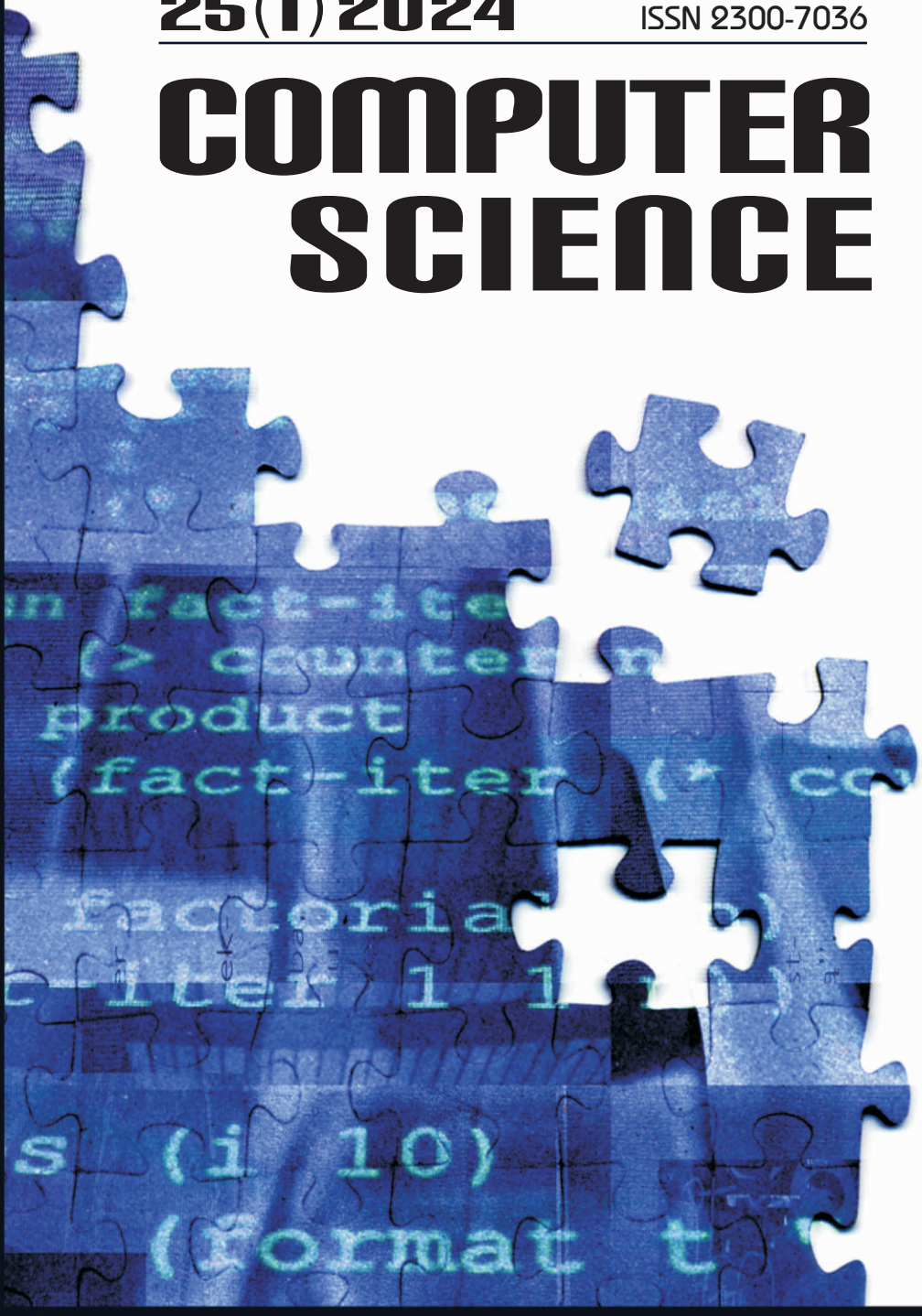
25(1)2024 ISSN 2300-7036

COMPUTER SCIENCE



QUARTERLY OF AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

AGH UNIVERSITY PRESS KRAKOW 2024

25(1) 2024 ISSN 2300-7036

COMPUTER SCIENCE

AGH UNIVERSITY PRESS KRAKOW 2024

COMPUTER SCIENCE is published by AGH University Press, Krakow, Poland.

The papers presented in COMPUTER SCIENCE have been accepted by the reviewers selected by the editors of the journal.

Head of Publishing of AGH University Press
Jan Sas

Technical Editor
Magdalena Grzech

Ghostwriting prevention
Łukasz Faber

Statistical Correction
Anna Barańska

Linguistic Correction
Bret Spainhour

Cover Design
Anna Sadowska

Typesetting and Desktop Publishing
Marek Karkula

ISSN 2300-7036
DOI: https://doi.org/10.7494/csci

Wydawnictwa AGH (AGH University Press)
al. A. Mickiewicza 30, 30-059 Kraków, Poland
tel. +48 12 617 32 28, +48 12 636 40 38
e-mail: redakcja@wydawnictwoagh.pl
https://www.wydawnictwa.agh.edu.pl

Computer Science • 25(1) 2024 https://doi.org/10.7494/csci.2024.25.1.6153

Mariusz Flasiński

A SURVEY ON SYNTACTIC PATTERN RECOGNITION METHODS IN BIOINFORMATICS

Abstract *Formal tools and models of syntactic pattern recognition which are used in bioinformatics are introduced and characterized in the paper. They include, among others: stochastic (string) grammars and automata, hidden Markov models, programmed grammars, attributed grammars, stochastic tree grammars, Tree Adjoining Grammars (TAGs), algebraic dynamic programming, NLC- and NCE-type graph grammars, and algebraic graph transformation systems. The survey of applications of these formal tools and models in bioinformatics is presented.*

Keywords bioinformatics, syntactic pattern recognition, generative grammar, hidden Markov model (HMM)

Citation Computer Science 25(1) 2024: 5–42

1. Introduction

There are two main subareas in pattern recognition: the decision-theoretic subarea [43, 114], including the neural network-based approach [156, 167], and the syntactic/structural one. The latter subarea can be divided, in turn, into the structural approach [25] and syntactic pattern recognition, SPR [61, 70]. In syntactic pattern recognition, a pattern takes the form of a string, a tree or a graph and a set of (structural) patterns is treated as a formal language. Then, a generative grammar is defined as a generator of this language and a syntax analyzer (formal automaton) is constructed for recognizing and/or interpreting of structural patterns. There are three groups of syntactic pattern recognition models depending on a type of a structure considered, namely: string-based models, tree-based models and graph-based models. We use this taxonomy for the presentation of syntactic pattern recognition methods in the paper. From the methodological point of view, syntactic pattern recognition is preferred if patterns considered are structural, a recognition process is multilevel and hierarchy-oriented and a structure-based interpretation is required [61].

As we will see in the next section, there are important issues considered in bioinformatics that can be characterized with the methodological requirements mentioned above. Indeed, syntactic pattern recognition has delivered formal models for recognizing and interpreting structural patterns in bioinformatics from the very beginning. In fact, the first SPR application took place in bioinformatics in the early 1960s. (The term *bioinformatics* had not yet been coined.) This was the development of the FIDAC system for karyotype analysis by Robert S. Ledley and his collaborators [120, 121].

The research areas and important problems of bioinformatics in the context of syntactic pattern recognition methods are introduced in Section 2. The basic formal tools and models of syntactic pattern recognition which are used in bioinformatics are characterized in the third section. It allows us to refer to these models in Section 4, in which the survey of syntactic pattern recognition applications in bioinformatics is presented. The last section contains conclusions.

2. Issues of bioinformatics and syntactic pattern recognition

There are three basic formal tools in syntactic pattern recognition: a (generative) grammar, a syntax analyzer (formal automaton, parser) and a language inference (induction) algorithm [61]. A grammar is a formal tool for the generating of a set of strings/sequences (trees, graphs) which is treated here as a formal language. Thus, the grammar models sequences (structures) *via* their generation. On the other hand, a syntax analyzer (automaton) is a formal tool for the recognition/classification of a set of sequences (structures). Thus, it models sequences (structures) *via* their analysis. In any case, both formal tools are used for the modeling of sequences/structures in order to better understand their structural properties. The typical applications of these formal tools in bioinformatics include: finding a subsequence/substructure that

relates to important features/functions in the whole sequence/structure, aligning sequences, predicting sequences on the basis of specific features, modeling a higher-level structure on the basis of a lower-level one (e.g. RNA tertiary structure on the basis of its secondary structure) and the like. A language inference (induction) algorithm is an algorithm which generates (automatically) a grammar or a syntax analyzer (automaton) on the basis of a sample of sequences (structures). In fact, it is a learning formal tool, i.e. it learns a model (represented by a grammar/automaton) on the basis of examples. For example, a task of this formal tool can be defined in the following way. Given a set of biological sequences, construct a stochastic automaton (or a grammar) that models these sequences.

Bioinformatics applies (and sometimes develops) models of computer science in order to better understand biological processes. These models and software systems constructed on their bases are especially useful when the data sets to be analyzed and interpreted are complex and large. The main research areas of bioinformatics include:

- *sequence analysis*,
- *structural bioinformatics*,
- *gene and protein expression*,
- *analysis of cellular organization*,
- *network and systems biology*.

Syntactic pattern recognition models have been applied especially in the first three areas. We will characterize them in a general way by identifying their main problems, since we refer to these problems in Section 4 which contains the survey of syntactic pattern recognition applications in bioinformatics.

Sequential structures are structures which are the most frequently considered in bioinformatics. Their constitute the primary structures of DNA, RNA and proteins. For example, the primary structure of the form of the amino acids' sequence of the ubiquitin protein is shown in Figure 1a. *Sequence analysis* consists in the analysis of DNA, RNA or protein sequence in order to understand their features, biological function or evolution. Its main issues involve, among others: a sequence alignment, a sequence assembly, and a gene prediction. A sequence alignment in bioinformatics is a way of arranging sequences (DNA, RNA, protein) in order to identify regions of similarity which may result from structural, functional or evolutionary relations. There are two issues here: a pairwise sequence alignment (an analysis of two sequences) and a multiple sequence alignment (an analysis of more than two sequences at a time). A sequence assembly consists in aligning and merging of fragments that belong to a longer sequence in order to obtain the original sequence. A gene prediction consists in finding the parts of genomic DNA that encode genes, mainly by identifying the stop and start regions of genes (which is called a gene annotation). Since sequence analysis problems concern an identification, recognition and/or interpretation of sequence, string-based models are the most convenient formal tools in this case. In Section 3.1 we present these models, including: stochastic grammars, stochastic automata, hidden Markov models, programmed grammars, and attributed grammars.

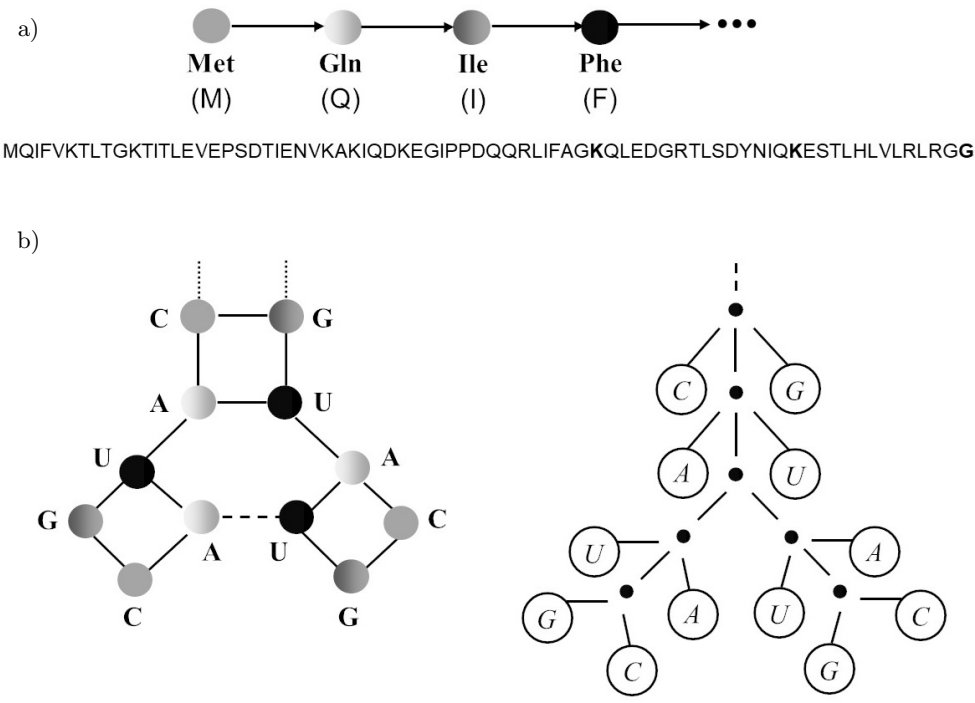


Figure 1. The beginning of the primary structure (the sequence of amino acids) of the ubiquitin protein (M stands for methionine (Met), Q stands for glutamine (Gln), I stands for isoleucine (Ile), F stands for phenylalaline (Phe) etc.) and its complete string representation (a). The exemplary part of the secondary structure of RNA (the branched RNA structure, and its tree representation generated by Tree Adjoining Grammar (b). Adapted from: [61]

Structural bioinformatics involves the analysis and prediction of higher-level, three-dimensional structure of proteins, RNA, and DNA. For example, a part of the secondary structure of RNA (the branched RNA structure, and its tree representation is shown in Figure 1b, whereas the graph structure of the nucleobase cytosine used in the modeling of the tertiary structure of RNA is shown in Figure 2. In case of proteins four structural levels are identified: the primary level that can be represented by sequences and three higher levels (secondary, tertiary, and quaternary) that are usually represented by trees or graphs. Protein structure prediction is one of the most important issues in structural bioinformatics, since the structure of a protein relates to its function. Therefore, the problem is crucial for medicine (drug design) as well as for biotechnology (novel enzymes design). It can be defined as the prediction of the secondary level- and tertiary level-structure on the basis of the primary level-(sequential) structure. The structures of RNA and DNA are represented by trees or

graphs in bioinformatics. Therefore, tree-based and graph-based models are used in this case. In Section 3.2 we present stochastic tree grammars, Tree Adjoining Grammars and algebraic dynamic programming, whereas in Section 3.3 we introduce NLC- and NCE-type graph grammars and algebraic graph transformation systems.

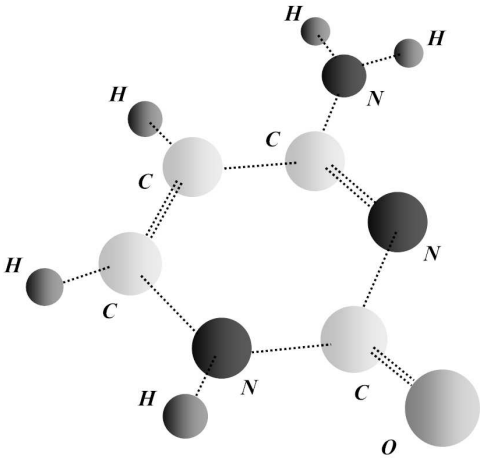


Figure 2. The graph structure of the nucleobase cytosine – a building block for the modeling of the tertiary structure of RNA

Gene and protein expression area studies three main issues: an analysis of a gene expression, a gene regulation, and an analysis of protein expression. This area contributes to medicine, pharmacy, and agriculture considerably. A gene expression consists in affecting a phenotype by information from a gene. This information is used in the synthesis of a functional gene product (RNA, protein). A gene regulation, in turn, is a process of the increasing/decreasing of the production of gene products by cells as a result of the appearing of some signal. String-based models of syntactic pattern recognition are used in the area of gene and protein expression.

3. Basic formal tools of syntactic pattern recognition for bioinformatics

Basic definitions and characteristics of main classes of grammars and automata used for bioinformatics are contained in this section. The string-, tree- and graph-based models are presented in the succeeding subsections.

3.1. String-based models

The generative power of grammars (and the discriminative power of the corresponding automata) of the standard Chomsky model is sometimes too small

for their effective use in the real-world applications. Therefore, a variety of enhanced grammars and automata have been defined to solve this problem [61]. The most useful approaches include: stochastic grammars/automata [68], fuzzy grammars/automata [207], error-correcting automata [192], hidden Markov models [11], and other enhanced models, e.g., programmed grammars [160], attributed grammars [113], and vague languages/multi-derivational parsing [66].

Computational biologists usually reason in the presence of uncertainty, because many facts are missing and often data are noisy. In order to handle this problem, probabilistic models, e.g. Bayesian inference, Markov Random Fields, variational methods, Bayesian networks etc., are applied in bioinformatics [9, 44]. The sequence analysis tasks of modeling, aligning, predicting etc. which have been discussed in the previous section, are of the probabilistic nature as well. Therefore, enhanced probabilistic formal tools of syntactic pattern recognition are often used in bioinformatics. Let us begin their presentation with stochastic regular grammars [17, 68, 72, 85, 163].

Definition 1. *A stochastic regular grammar is a quadruple*

$$G = (\Sigma_N, \Sigma_T, P, S), \text{ where}$$

Σ_N *is a set of nonterminal symbols,*
Σ_T *is a set of terminal symbols,*
P *is a set of stochastic productions of the form:*

$$A_i \xrightarrow{p_{ij}} \gamma_{ij}, \quad i = 1, \ldots, n, \ j = 1, \ldots, m_i,$$

in which $A_i \in \Sigma_N$, $\gamma_{ij} \in \Sigma_T \cup \Sigma_T\Sigma_N$, p_{ij} *is the probability related to the application of the production such that*

$$0 < p_{ij} \leq 1 \quad , \quad \sum_{j=1}^{m_i} p_{ij} = 1 \ ,$$

S *is the start symbol (axiom),* $S \in \Sigma_N$. □

Thus, a stochastic grammar is a standard (Chomsky) grammar such that probabilities have been ascribed to productions. In this case a derivation definition has to be modified slightly. Let the string θ be derived directly from the string β, denoted $\beta \overset{p_{ij}}{\Longrightarrow} \theta$, as the result of applying the production $A_i \xrightarrow{p_{ij}} \gamma_{ij}$.

We say that α_1 derives α_r with the probability $p = \prod_{k=1}^{r} p_k$, denoted $\alpha_1 \overset{p}{\underset{*}{\Longrightarrow}} \alpha_r$ iff there exists the following sequence of derivational steps

$$\alpha_k \overset{p_k}{\Longrightarrow} \alpha_{k+1} \ , \ k = 1, \ldots, r-1 \ .$$

The stochastic language generated by the grammar is defined as follows.

Definition 2. *The language generated by the stochastic regular grammar* $G = (\Sigma_N, \Sigma_T, P, S)$ *is the set*

$$L(G) = \{(\phi, p(\phi)) : \phi \in \Sigma_T^*, \quad S \overset{p_v}{\underset{*}{\Longrightarrow}} \phi, \ v = 1, \ldots, s, \quad p(\phi) = \sum_{v=1}^{s} p_v\},$$

where s *is the number of all the different derivations of* ϕ *from* S *and* p_v *is the probability of the* v*th derivation of* ϕ. □

For stochastic grammars of various types (i.e., regular, context-free etc.), the corresponding classes of stochastic automata (i.e., finite-state, pushdown etc.) are defined. They differ from their standard counterparts in the ascribing of probabilities to their transitions. Thus, stochastic finite-state automaton (FSA) is defined in the following way [72, 153, 194].

Definition 3. *A stochastic finite-state automaton is a quintuple*

$$A = (Q, \Sigma_T, \Pi, \pi_0, \pi_F), \text{ where}$$

Q *is a set of* n *states,*
Σ_T *is a finite set of input symbols,*
Π *is a mapping of* Σ_T *into the set of* $n \times n$ *stochastic state-transition matrices such that*

$$\Pi(a) = [\pi_{ij}(a)]_{n\times n} \ , \ \pi_{ij} \geq 0 \ , \ \sum_{j=1}^{n} \pi_{ij} = 1 \ , \ i = 1, \ldots, n \ ,$$

where $\pi_{ij}(a)$ *is the probability of the transition from state* q_i *to state* q_j *when the symbol* a *has been read,*
π_0 *is an* n*-dimensional row vector representing the initial state distribution such that its first component is equal to 1 and the remaining components are equal to 0,*
π_F *is an* n*-dimensional column vector such that its* k*th component is equal to 1 if* q_k *is the final state and 0 otherwise.* □

A stochastic FSA corresponds to a Markov chain defined in the theory of stochastic processes. In both cases, i.e. a stochastic FSA and a Markov chain, we assume that we the probabilities for sequences of observable events are known. (That is, a stochastic process is observable which means that any transition between two states in a stochastic FSA is related to one symbol.) In bioinformatics, however, such an assumption is too strong, i.e. the events we are interested in can be not observable directly. In the theory of syntactic pattern recognition we use an enhanced model of a stochastic FSA, namely *hidden Markov model, HMM* in such a case. (HMMs were firstly applied in the 1960s in the field of Natural Language Processing.) Then, in case of bioinformatics, a hidden Markov model transits through a series of "hidden" states, modeling a biological sequence (denoting e.g. a protein) by *emitting* succeeding terminal symbols (corresponding to e.g. amino acids). (In the case of *HMMs* we say that a terminal symbol is *emitted* instead of saying that it is *generated/read.*) Any state of a HMM does not have to be related one-to-one to the event observed (as in case of stochastic finite-state automata), but the probability distribution for a set of terminal symbols is defined for each state independently. Let us formalize our considerations with the following definition [11, 138].

Definition 4. *A hidden Markov (HMM) model is a quintuple*

$$HMM = (Q, \Sigma_T, \Pi, E, \pi_0), \text{ where}$$

$Q = \{q_1, q_2, \ldots, q_N\}$ *is a set of* N *states,*
$\Sigma_T = \{a_1, a_2, \ldots, a_M\}$ *is a finite set of* M *symbols,*
$\Pi : Q \times Q \to \mathbb{R}_{\geq 0}$ *is the state-transition probability distribution,*
$E : Q \times \Sigma_T \to \mathbb{R}_{\geq 0}$ *is the state-based symbol emission probability distribution,*

$\pi_0 = [\pi(1), \pi(2), \ldots, \pi(N)]$ *is the initial state distribution vector, and the following conditions hold:*

$$\forall q' \in Q \quad \sum_{q'' \in Q} \Pi(q', q'') = 1 \quad , \quad \sum_{a \in \Sigma_T} E(q', a) = 1 \quad , \quad \sum_{i=1}^{N} \pi(i) = 1 \ . \ \square$$

$\Pi(q_i, q_j) = \pi_{ij}, i, j = 1, \ldots, N$ is the probability of the transition from state q_i to state q_j. $E(q_j, a_m) = e_j(a_m), j = 1, \ldots, N, \ m = 1, \ldots, M$ is the probability of emitting a_m in state q_j. $\pi(i), i = 1, \ldots, N$ is the probability the Markov chain starts in state q_i.

In the theory of syntactic pattern recognition, error-correcting automata [192] are applied in two cases [61]. Firstly, they are used, if we have to analyze distorted/deformed versions of structural representations of reference (template) patterns. Secondly, they can model the family of variant patterns belonging to the same category (class), yet differing from each other in some detailed structural features. Then, the "error"-transformations are defined for the string representations and the *expanded grammar* is constructed by adding "error"-productions which model these structural differences in the variant patterns. Finally, the error-correcting automaton that, apart from normal states, contains error-states (and error-transitions) is constructed. The biological sequences usually come in families. Then, the sequences which belong to the same family diverge from each other. For modeling protein family assignment, multiple sequence alignment, protein structure prediction, alignment segmentation, etc. hidden Markov models, presented above, also have been enhanced, by introducing the so-called, *profile hidden Markov models* [47, 87, 115].

In order to formalize our considerations, we introduce the notion of (error) string transformation [70, 125] as in [61].

Let there be given two strings $x, y \in \Sigma_T^*$. A transformation $\mathcal{F} : \Sigma_T^* \longmapsto \Sigma_T^*$ such that $y \in \mathcal{F}(x)$ is called a *string transformation*. The following string (error) transformations are defined:

- Substitution transformation $\mathcal{F}_S : \eta_1 a \eta_2 \stackrel{\mathcal{F}_S}{\longmapsto} \eta_1 b \eta_2 \ , \ a, b \in \Sigma_T, \ a \neq b, \ \eta_1, \eta_2 \in \Sigma_T^*$.
- Insertion transformation $\mathcal{F}_I : \eta_1 \eta_2 \stackrel{\mathcal{F}_I}{\longmapsto} \eta_1 a \eta_2 \ , \ a \in \Sigma_T, \ \eta_1, \eta_2 \in \Sigma_T^*$.
- Deletion transformation $\mathcal{F}_D : \eta_1 a \eta_2 \stackrel{\mathcal{F}_D}{\longmapsto} \eta_1 \eta_2 \ , \ a \in \Sigma_T, \ \eta_1, \eta_2 \in \Sigma_T^*$.

For defining profile hidden Markov models, only insertion and deletion transformations are used. A profile hidden Markov model is just a hidden Markov model such that three types of states, namely: (normal) match states, insert states (modeling insertion transformation), and delete states (modeling deletion transformations), are distinguished, and its generic structure is defined as it is shown in Figure 3. Summing up, profile hidden Markov models can be considered to be *error-correcting* hidden Markov models, as it understood in syntactic pattern recognition.

Two basic classes of Chomsky grammars are used in syntactic pattern recognition: (weaker) regular grammars and (stronger) context-free grammars (CFGs).

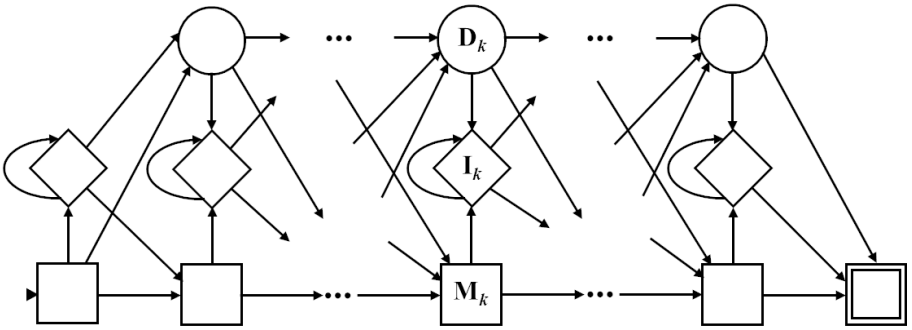


Figure 3. The generic structure of a profile HMM (Normal (match) states are represented with squares, insert states are represented with diamonds and delete states are represented with circles. The begin state is marked with small black triangle and the end state is marked with double square)

However, sometimes even CFGs are too weak if a generative power is concerned, i.e. a language (a set of sequential patterns) is too complex to be generated by any CFG. For example, a language considered can be context-sensitive (CSL). In bioinformatics such a problem arises quite frequently, e.g. in the case of some RNA pseudoknotted structures [44,157]. In syntactic pattern recognition, such structures can be viewed as crossing interactions which can be modeled with the help of the copy language L_c that is of the form $L_c = \{ww : w \in \Sigma_T^*\}$. However, L_c is the context-sensitive language generated by context-sensitive grammars (CSGs). The problem is that context-sensitive grammars are inefficient computationally and therefore they are not used in practical applications. Such a problem is effectively solved in syntactic pattern recognition by defining various classes of *enhanced CFGs* which can generate certain context-sensitive languages. (For a review of such enhanced grammars, see [61].) Programmed CFGs, introduced in [160], are one of the most popular enhanced CFGs. Let us present their definition.

Definition 5. *A programmed context-free grammar is a quintuple*

$$G = (\Sigma_N, \Sigma_T, J, P, S), \text{ where}$$

Σ_N *is a set of nonterminal symbols,*
Σ_T *is a set of terminal symbols,*
J *is a set of production labels,*
P *is a finite set of productions of the form:*

$$(r) \quad A \to \beta \quad S(U) \quad F(W), \text{ in which}$$

$A \to \beta$, $A \in \Sigma_N$, $\beta \in \Sigma^*$, *is called the core,* (r) *is the production label,* $r \in J$, $U \subset J$ *is the success field and* $W \subset J$ *is the failure field,*
S *is the start symbol (axiom),* $S \in \Sigma_N$. □

A derivation in a programmed CFG can be defined as follows. Firstly, the production labeled with (1) is applied. If any production is applied, then after its application

the next production is chosen from its success field U. Otherwise, the next production is chosen from the failure field W. Intuitively speaking, a programming mechanism allows us to control the choice of subsequent productions during a derivation, and this way to force the applying of some (desirable) productions in case a certain production has been applied before. For example, we can generate the (context-sensitive) copy language L_c with a programmed context-free grammar.

The extension of programmed CFGs, namely (dynamically programmed) DPLL(k) grammars have been defined in [62]. They are more efficient computationally, i.e. their syntax analyzer is only of the $\mathcal{O}(n^2)$ time complexity. Their extensions to the error-correcting model and the stochastic model have been defined as well [61]. DPLL(k) grammars can generate such typical (complex) context sensitive-languages like, e.g., $L_1 = \{a^n b^n c^n : n \geq 0\}$, $L_2 = \{a^n b^m c^n d^m : n, m \geq 0\}$ [61].

If symbolic/structural information on structural patterns that is represented by a formal language/grammar should be supplied with numerical information, then attribute grammars are used in syntactic pattern recognition. Such a use of numerical information can be required in the case of minimum-distance alignment or folding operations performed for biological sequences [123,171]. Let us introduce the following notions and definitions.

Let A_X denote the set of attributes of the symbol $X \in \Sigma$, $X_\bullet\alpha$ denote the attribute α of X, D_α denote the set of possible values for the attribute α.

Let $(p)\ \ X^0 \rightarrow X^1X^2 \ldots X^m$ be a production of a context-free grammar and $A^{(p)} = A_{X^0} \cup A_{X^1} \cup A_{X^2} \cup \ldots \cup A_{X^m}$. A *semantic rule* for the production (p) is an expression of the following form

$$\beta := f(\gamma_1, \gamma_2, \ldots, \gamma_k), \text{ where}$$

$\beta, \gamma_1, \gamma_2, \ldots, \gamma_k \in A^{(p)}$,
$f : D_{\gamma_1} \times D_{\gamma_2} \times \ldots \times D_{\gamma_k} \rightarrow D_\beta$ is a function. The set of semantic rules for the production (p) is denoted by $R^{(p)}$.

Now, we can present attributed context-free grammars as in [70,113].

Definition 6. *An attributed context-free grammar is a sextuple*

$$G = (\Sigma_N, \Sigma_T, P, S, A, R), \textit{ where}$$

Σ_N, Σ_T, P, S *are defined as for a context-free grammar,*
$A = \bigcup_{X \in \Sigma} A_X$ *is a finite set of attributes,*
$R = \bigcup_{p \in P} R^{(p)}$ *is a finite set of semantic rules.* □

Since values can be ascribed to attributes according to semantic rules related to productions (syntactic rules) of a grammar, the corresponding syntax analyzer can compute certain measures during succeeding steps of parsing. These measures can be, then, used for the evaluation of distances between analyzed sequences, which is very useful in case of operations performed for biological sequences mentioned above.

3.2. Tree-based models

As we have mentioned in Section 2, tree languages are used mainly for the analysis and prediction of higher-level structures. It includes: the prediction of protein

secondary structures, the prediction of RNA secondary structures (cf. Figure 1b in Section 2), and the prediction of tertiary interactions over pseudoknots for RNA secondary structures. In this section, the most popular tree-based models used in these tasks are presented subsequently, i.e.: (stochastic) tree grammars, Tree Adjoining Grammars, and Algebraic Dynamic Programming.

We introduce the notions concerning tree structures [18, 34, 70, 74] as in [61].

Let $\mathcal{U} = (\mathbb{N}_+, \bullet, \lambda)$, where $\mathbb{N}_+$ is the set of positive integers, $\bullet$ is the operation, λ is the identity, be the free monoid. The partial ordering $\leq$ on $\mathcal{U}$ is defined as follows. $x \leq y,\ x, y \in \mathcal{U}$ iff there exists $z \in \mathcal{U}$ such that $x \bullet z = y$. x and y are incomparable iff $x \not\leq y$ and $y \not\leq x$. $\mathcal{U}$ is called the *Gorn universal tree domain*.

A subset $D \subset \mathcal{U}$ is a *tree domain* iff for all $x, y \in \mathcal{U}$ and all $i, j \in \mathbb{N}_+$ the following conditions are satisfied: (1) if $x \bullet y \in D$ then $x \in D$ and (2) if $x \bullet j \in D$ and $i \leq j$ then $x \bullet i \in D$. The *root* is represented by λ. The *leaves* are the nodes which are maximal with respect to $\leq$. A tree node which is not a leaf is called an *internal node*.

Let $\mathbb{N}$ be the set of nonnegative integers, A be a finite subset of $\mathbb{N}$. A *ranked alphabet* is a pair (Σ, r), where Σ is a finite alphabet, $r : \Sigma \to 2^A$ is a rank multi-valued mapping. $n \in r(a), a \in \Sigma$ is called the rank of a. We denote $\Sigma_n = \{a : n \in r(a)\}$.

A *tree* over (Σ, r) is a function $t : D \to \Sigma$, D is a tree domain, such that: (1) $t(x) \in \Sigma_0$, if x is a leaf in D and (2) $t(x) \in \Sigma_n$, where $n = max\{i \in \mathbb{N}_+ : x \bullet i \in D\}$, otherwise. The domain of a tree t is denoted by D_t. The set of all finite trees over Σ is denoted by T_Σ. A *node* is a pair $(x, a) \in D \times \Sigma$. The *frontier* of t is the sequence of its leaves.

Let $t \in T_\Sigma$ and $x \in D_t$. The subtree of t at x, denoted t/x, is defined by the function which is the set of pairs $\{(y, a) : (x \bullet y, a) \in t,\ a \in \Sigma\}$.

Now, we can introduce the definition of (expansive) stochastic regular tree grammars [16, 69, 70, 132].

Definition 7. *An (expansive) stochastic regular tree grammar over* (Σ_T, r) *is a quintuple*

$$G = (\Sigma_N, \Sigma_T, r, P, S), \text{ where}$$

Σ_N *is a finite set of nonterminal symbols,*
(Σ_T, r) *is a ranked alphabet of terminal symbols,* $\Sigma_N \cap \Sigma_T = \emptyset$, $\Sigma = \Sigma_N \cup \Sigma_T$,
P *is a set of productions of the form:*

$$A_i \xrightarrow{p_{ij}} t_{ij}, \quad i = 1, \ldots, n,\ j = 1, \ldots, m_i,$$

in which $A_i \in \Sigma_N$, $t_{ij} \in T_\Sigma$ *is a tree which either consists of a terminal root and its nonterminal children or consists of a terminal node,* p_{ij} *is the probability related to the application of the production such that*

$$0 < p_{ij} \leq 1 \quad , \quad \sum_{j=1}^{m_i} p_{ij} = 1 \ ,$$

$S \in \Sigma_N$ *is the start symbol.* □

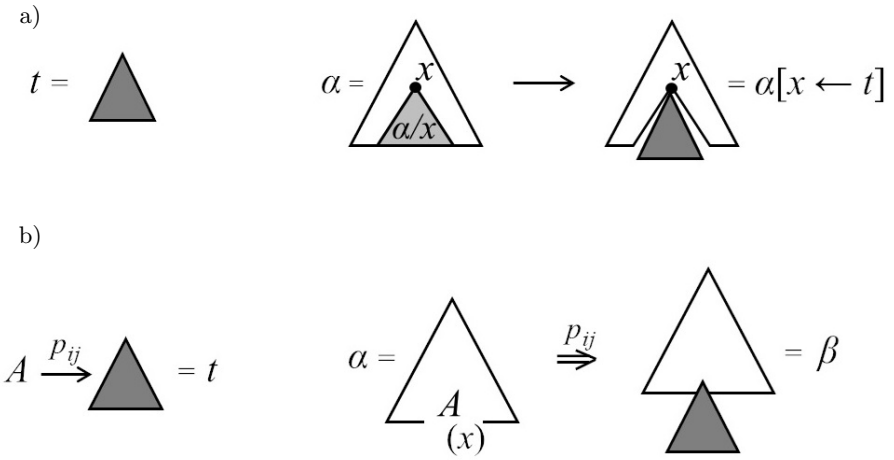


Figure 4. Replacement of a subtree (a) and derivation in stochastic regular tree grammar (b)

The definition of standard (non-stochastic) tree grammar can be obtained by removing the probabilities in Definition 7.

A derivation step is introduced as a kind of more general operation of subtree replacement. The *replacement* of the subtree α/x by t, denoted $\alpha[x \leftarrow t]$, is the tree defined by the function which is the set of pairs (see Fig. 4a)

$$\{(y, \alpha(y)) : y \in D_\alpha \ , \ x \text{ is not a prefix of } y\} \cup \{(x \bullet z, t(z)) : z \in D_t\}.$$

Let $\alpha, \beta \in T_\Sigma$ and $x \in D_\alpha$. α directly derives β with the probability p_{ij} in G, denoted $\alpha \overset{p_{ij}}{\Longrightarrow} \beta$, iff there exists $A \overset{p_{ij}}{\rightarrow} t \in P$ such that $\alpha(x) = A$ and $\beta = \alpha[x \leftarrow t]$ (see Fig. 4 (b)). The stochastic tree language generated by the stochastic tree grammar G is defined in an analogous way as the stochastic string language (cf. Definition 2).

Now, we present Tree Adjoining Grammars (*TAGs*) [100–102] according to [61].

Definition 8. *A Tree Adjoining Grammar, TAG, is a quintuple*

$$G = (\Sigma_N, \Sigma_T, S, I, A), \text{ where}$$

Σ_N *is a finite set of nonterminal symbols,*
Σ_T *is a finite set of terminal symbols,* $\Sigma_N \cap \Sigma_T = \emptyset$, $\Sigma = \Sigma_N \cup \Sigma_T$,
$S \in \Sigma_N$ *is the initial symbol,*
I *is a finite set of initial trees such that for any* $\alpha \in I$ *the internal nodes of* α *are labelled by nonterminals and leaves are labeled by terminals or nonterminals; nonterminal leaves of* α *are marked for the substitution operation with a special symbol* $\downarrow$,
A *is a finite set of auxiliary trees such that for any* $\beta \in A$ *the internal nodes of* β *are labelled by nonterminals and leaves are labeled by terminals or nonterminals; nonterminal leaves of* β *are marked for substitution except for one node, called the foot node; the foot node has the same label as the root of* β*; the foot node is marked for the adjoining operation with a special symbol* $*$. □

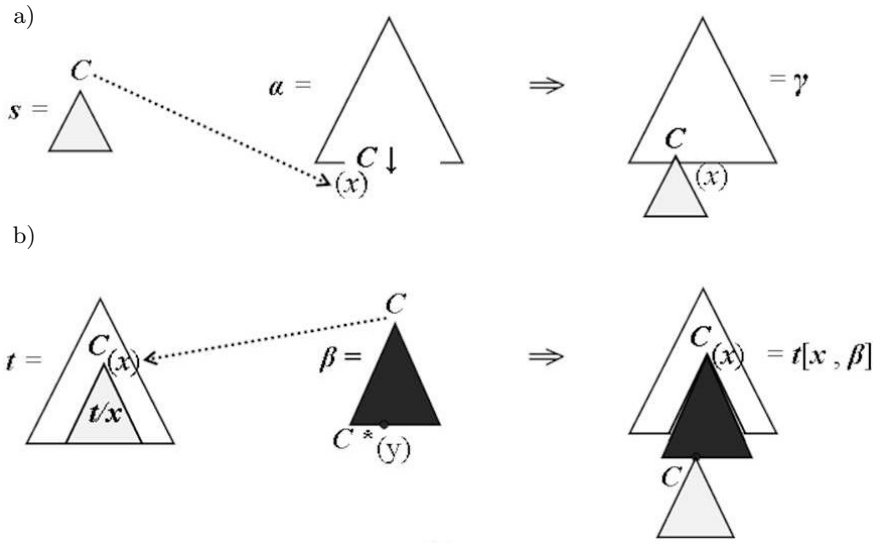


Figure 5. Substitution in TAG (a), adjoining in TAG (b)

The scheme of substitution operation is shown in Figure 5a. A nonterminal leaf marked $\downarrow$ of a derived tree is replaced with some tree s derived from an initial tree. The replaced node should have the same label as the root of s.

The scheme of adjoining operation is shown in Figure 5b. An auxiliary tree β is inserted into an internal node having the address x of a derived tree t. The node of t having the address x should have the same label as the root of β. The subtree t/x is attached to the foot node of β which is marked with $*$.

Let $\theta, \gamma \in T_\Sigma$. θ directly derives γ in G, denoted $\theta \underset{G}{\Longrightarrow} \gamma$, iff either $\gamma = \theta[x, \beta]$, $x \in D_\theta$, $\beta \in A$ or γ results from the application of a substitution operation to θ.

The reflexive and transitive closure of the relation $\underset{G}{\Longrightarrow}$ is denoted with $\overset{*}{\underset{G}{\Longrightarrow}}$. If $\theta \overset{*}{\underset{G}{\Longrightarrow}} \gamma$, then γ is called a *derived tree of* θ. The set of all derived trees of θ is denoted with $DT(\theta)$.

Now, we can define the tree language generated by TAG G.

Definition 9. *The tree language generated by TAG G is the set*

$$T(G) = \{\gamma \in T_\Sigma : \gamma \in DT(\theta),\ \theta \in I,\ \theta(\lambda) = S,\ \textit{and}\ Y(\gamma) \in \Sigma_T^*\}.\ \square$$

At the end of this section, we present the novel efficient approach of *Algebraic Dynamic Programming*, (*ADP*) [76, 78, 79, 81, 82, 166] which has been developed in bioinformatics. This approach is based on the (well-known in computer science) paradigm of *dynamic programming* which is a generic model of the constructing of efficient algorithms for complex problems which, by definition, involve the searching of a space of exponential size (that is inefficient computationally). The paradigm consists in breaking a complex problem into simpler subproblems recursively (in case these subproblems are shared) which allows one to search the space in polynomial time [12].

Dynamic programming algorithms are widely used in bioinformatics, including: optimal global alignment, local alignment, repeated matching, overlap matching, etc. [44].

Algebraic Dynamic Programming is a systematic methodology of the constructing of dynamic programming algorithms. Two main phases are defined in the methodology: the recognition phase and the evaluation phase.

During the recognition phase a *yield grammar* is used. The concept of *yield* has been introduced for Tree Adjoining Grammars, presented above. Let us introduce this concept according to [61].

Let us define the *yield mapping* $Y : T_\Sigma \longrightarrow \Sigma_{T,0}^*$ in the following way.
(1) If $a \in \Sigma_{T,0}$ then $Y(a) = a$.
(2) If $a \in \Sigma_{T,n}$, $k > 0$ and $t_1, t_2, \ldots, t_n \in T_\Sigma$ then

$$Y(a(t_1 t_2 \ldots t_n)) = Y(t_1) \ \cdot \ Y(t_2) \ \cdot \ \ldots \ \cdot \ Y(t_n)) \ ,$$

where $\cdot$ is the concatenation operation.

Thus, yield mapping delivers the sequence of the labels of the frontier nodes (i.e. the leaves), writing them from left to right.

For tree grammars we can define the tree language generated by them, as it has been made by Definition 9 for Tree Adjoining Grammars. On the other hand, we can also define the string language generated by them in the following way.

Definition 10. *The string language generated by TAG G is the set*

$$L(G) = \{v : v = Y(\gamma), \ \gamma \in T(G)\}. \ \square$$

In this case the strings defined by the terminal labels of the frontiers of the derived trees are treated as the words of this (string) language. Then, we say that G is the yield grammar. In fact, Tree Adjoining Grammars have been introduced in syntactic pattern recognition for generating *enhanced context-free (string) grammars* that has been discussed in the previous section. The search space of the problem considered is described by the yield grammar.

During the evaluation phase, the so-called *evaluation Σ-algebra* (an *interpretation*, as it is understood in algebraic semantics) is used to comprise the aspects relevant to the objective assumed, independently of the description of the search space by the yield grammar. This way the dynamic programming algorithms can be developed on a more abstract level than in the standard dynamic programming approach. *Algebraic Dynamic Programming* methodology has been successfully used, among others, for sequence alignment and RNA folding.

3.3. Graph-based models

Graph grammars are the strongest generative formalism in syntactic pattern recognition [61, 70, 145], because every kind of relation among the elements of a structure can be defined. Due to their big generative power, graph grammars have been used for such complex problems in bioinformatics as, e.g.: modeling RNA tertiary structure motifs, modeling RNA folding, modeling protein structures, genetic regulation, analyzing metabolic networks.

There are many classes of graph grammars [51]. In this subsection we present three classes which, on one hand, are classic in the theory of graph grammars and, on the other hand, are applied in bioinformatics. They include: Node Label Controlled (NLC) graph grammars, Neighborhood-Controlled Embedding (NCE) graph grammars, and algebraic (DPO) graph transformation systems.

As we have discussed in Section 3.1, the application of a certain class of a generative grammar is conditioned by the computational efficiency of the corresponding type of a syntax analyzer. In case of graph grammars this problem is especially crucial, because the research into the efficiency of graph parsing revealed a hard membership problem, PSPACE-complete or NP-complete, for graph grammars [20, 95, 181, 195]. (The reasons for the intractability of this problem were identified in [58, 61]). Fortunately, for the graph grammars of the Node Label Controlled (NLC) class (*edNLC* graph grammars), efficient, $\mathcal{O}(n^2)$, top-down ($ETPL(k)$) and bottom-up ($ETPR(k)$) syntax analyzers [54–56, 60, 61] as well as an efficient inference algorithm [59] have been defined. In result $ETPL(k)/ETPR(k)$ subclasses of NLC grammars could have been applied, among others, for scene analysis [54, 56], CAD/CAM integration [57], Polish Sign Language recognition [65]. The error-correcting $ETPL(k)$ syntax analyzer and its attributed version have been used for the recognition of vague/variant patterns [55, 64]. Stochastic $ETPL(k)$ grammars were applied for manufacturing quality control [61], and attributed programmed $ETPL(k)$ grammars – for process monitoring and control [63].

Let us introduce the notions concerning this class of graph grammars according to [94, 95, 97].

A directed node- and edge-labeled graph, *EDG* graph, over Σ and Γ is a quintuple $H = (V, E, \Sigma, \Gamma, \phi)$, where V is a finite, non-empty set of nodes, Σ is a finite, non-empty set of node labels, Γ is a finite, non-empty set of edge labels, E is a set of edges of the form (v, γ, w), in which $v, w \in V, \gamma \in \Gamma$, and $\phi : V \to \Sigma$ is a node-labeling function.

The family of the *EDG* graphs over Σ and Γ is denoted by $EDG_{\Sigma,\Gamma}$. The components V, E, ϕ of a graph H are sometimes denoted with V_H, E_H, ϕ_H.

Let $A = (V_A, E_A, \Sigma, \Gamma, \phi_A)$, $B = (V_B, E_B, \Sigma, \Gamma, \phi_B)$ and $C = (V_C, E_C, \Sigma, \Gamma, \phi_C)$ be *EDG* graphs. An isomorphism from A onto B is a bijective function h from V_A onto V_B such that

$$\phi_B \circ h = \phi_A \quad \textit{and} \quad E_B = \{(h(v), \gamma, h(w)) : (v, \gamma, w) \in E_A\}.$$

We say that A is *isomorphic to* B, and denote this with $A \cong B$.

Definition 11. *An edge-labeled directed Node Label Controlled, edNLC, graph grammar is a quintuple*

$$G = (\Sigma, \Sigma_T, \Gamma, P, Z), \textit{ where}$$

Σ *is a finite, non-empty set of node labels,*
$\Sigma_T \subseteq \Sigma$ *is a set of terminal node labels,*

Γ *is a finite, non-empty set of edge labels,*
P *is a finite set of productions of the form* (l, D, C)*, in which*
$l \in \Sigma \setminus \Sigma_T$, $D \in EDG_{\Sigma,\Gamma}$,
$C : \Gamma \times \{in, out\} \rightarrow 2^{\Sigma \times \Sigma \times \Gamma \times \{in,out\}}$ *is the embedding transformation,*
$Z \in EDG_{\Sigma,\Gamma}$ *is the start graph called the axiom.* $\square$

We have presented definitions for languages which consist of directed node- and edge-labeled graphs. If we use undirected node- and edge-labeled graphs, we denote the family of such graphs by $EG_{\Sigma,\Gamma}$ and the class of the corresponding graph grammars by *eNLC*. If we use undirected node-labeled graphs, we denote the family of such graphs by G_Σ and the class of the corresponding graph grammars by *NLC*. In both cases, the corresponding definitions are just simplified with relation to the definitions of *EDG* graphs and *edNLC* grammars (The same holds for definitions presented below).

A direct derivational step in edNLC graph grammars is defined as follows.

Definition 12. *Let* $G = (\Sigma, \Sigma_T, \Gamma, P, Z)$ *be an edNLC graph grammar.*
Let $H, \overline{H} \in EDG_{\Sigma,\Gamma}$*. Then* H *directly derives* $\overline{H}$ *in* G*, denoted by* $H \underset{G}{\Longrightarrow} \overline{H}$*, if there exists a node* $v \in V_H$ *and a production* (l, D, C) *in* P *such that the following holds.*

(a) $l = \phi_H(v)$.

(b) There exists an isomorphism from $\overline{H}$ *onto the graph* X *in* $EDG_{\Sigma,\Gamma}$ *constructed as follows. Let* $\overline{D}$ *be a graph isomorphic to* D *such that* $V_H \cap V_{\overline{D}} = \emptyset$ *and let* h *be an isomorphism from* D *onto* $\overline{D}$*. Then*

$$X = (V_X, E_X, \Sigma, \Gamma, \phi_X), \text{ where}$$

$V_X = (V_H \setminus \{v\}) \cup V_{\overline{D}}$,

$$\phi_X(y) = \begin{cases} \phi_H(y), & \text{if } y \in V_H \setminus \{v\}, \\ \phi_{\overline{D}}(y), & \text{if } y \in V_{\overline{D}} , \end{cases}$$

$E_X = (E_H \setminus \{(n, \gamma, m) : n = v \text{ or } m = v\}) \cup$
$\cup \{(n, \gamma, m) : n \in V_{\overline{D}}$, $m \in V_{X \setminus \overline{D}}$ *and there exists an edge* $(m, \lambda, v) \in E_H$ *such that* $(\phi_X(n), \phi_X(m), \gamma, out) \in C(\lambda, in)\} \cup$
$\cup \{(m, \gamma, n) : n \in V_{\overline{D}}$, $m \in V_{X \setminus \overline{D}}$ *and there exists an edge* $(m, \lambda, v) \in E_H$ *such that* $(\phi_X(n), \phi_X(m), \gamma, in) \in C(\lambda, in)\} \cup$
$\cup \{(n, \gamma, m) : n \in V_{\overline{D}}$, $m \in V_{X \setminus \overline{D}}$ *and there exists an edge* $(v, \lambda, m) \in E_H$ *such that* $(\phi_X(n), \phi_X(m), \gamma, out) \in C(\lambda, out)\} \cup$
$\cup \{(m, \gamma, n) : n \in V_{\overline{D}}$, $m \in V_{X \setminus \overline{D}}$ *and there exists an edge* $(v, \lambda, m) \in E_H$ *such that* $(\phi_X(n), \phi_X(m), \gamma, in) \in C(\lambda, out)\}$. $\square$

Since the definition of a derivation step for *edNLC* graph grammar is a little bit complicated, let us consider the following example. The start graph Z which a production is to be applied for, is shown in Figure 6a.

The left- and right-hand sides of a production to be applied are shown in Figure 6b. The embedding transformation of the production is defined in the following way.

(i) $C(\omega, out) = \{(r, X, \psi, in)\}$,

(ii) $C(\pi, in) = \{(d, t, \pi, in)\}$.

A derived graph h (the result of applying the production to the start graph Z) is shown in Figure 6c.

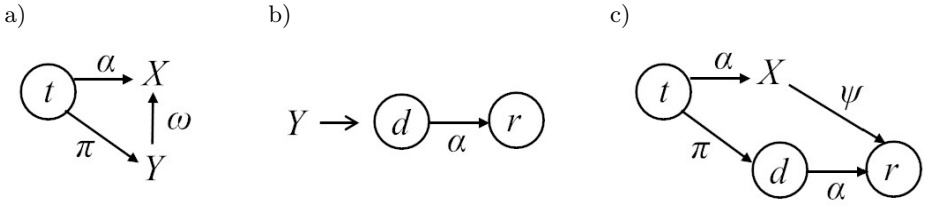


Figure 6. The start graph Z of the *edNLC* grammar G (a), a production of G (b) and the derived graph h (c)

The derivation step has two phases. During the first phase, the node labeled with Y of the graph Z is removed, and the graph of the right-hand side replaces the removed node. The transformed graph obtained by removing the node and its adjacent edges is called the rest graph. During the second phase, the embedding transformation is applied to connect some nodes of the right-hand side graph with the rest graph. The item **(i)** is interpreted as follows. Each edge labeled with ω and going *out* from the node corresponding to the left-hand side of a production, i.e. Y, has to be replaced by the edge: which connects the node of the graph of the right-hand side of the production and labeled with r with the node of the rest graph and labeled with X, is labeled with ψ, and comes *in* to the node r.

One can easily notice that the item **(ii)** just preserves the edge labeled with π.

(Indirect) derivations in the *edNLC* graph grammar G and the language generated by G are defined in an analogous way as for Chomsky (standard) grammars.

The class of *Neighborhood-Controlled Embedding* (*NCE*) *graph grammars* [96] is the extension (and enhancement) of the class of NLC graph grammars. The left-hand side of a production can be a graph (not only a nonterminal symbol). The embedding transformation uses node identifiers (not node labels) which allows us to distinguish various nodes having the same label.

Algebraic graph transformation systems – double pushout model (*DPO*) – were introduced in [52, 53]. Let us present the notions of: a production and a direct graph transformation in the DPO model according to [50].

A *production* in DPO is a triple $p = (L, K, R)$, where L is the left-hand side graph of p, R is the right-hand side graph of p, K is used for defining the gluing conditions.

The scheme of a DPO graph transformation is shown in Figure 7a. Let G be a graph which is to be transformed as the result of the application of a production

$p = (L, K, R)$. Let $L \setminus K$ denote the part of G which is to be removed from as the result of the application of p, $R \setminus K$ denote the part of G which is to be added to. A *direct graph transformation* with $p = (L, K, R)$ is performed in the following two steps.
(1) A match m of L in G is found such that m is structure-preserving. Then, all the nodes and edges which are matched with $L \setminus K$ are removed from G. (Let us note that m should satisfy a gluing condition, i.e. the gluing of $L \setminus K$ and D equals to G.) This step is depicted schematically in the part (1) of in Figure 7a.
(2) The graph D is glued with $R \setminus K$ in order to obtain the derived graph H, as it is depicted schematically in the part (2) of in Figure 7a. The graph K is used for gluing the nodes and edges which has been newly created into D. (It allows us to define the gluing points at which the right-hand side graph R is embedded into D.)

The example of the DPO graph transformation is shown in Figure 7b.

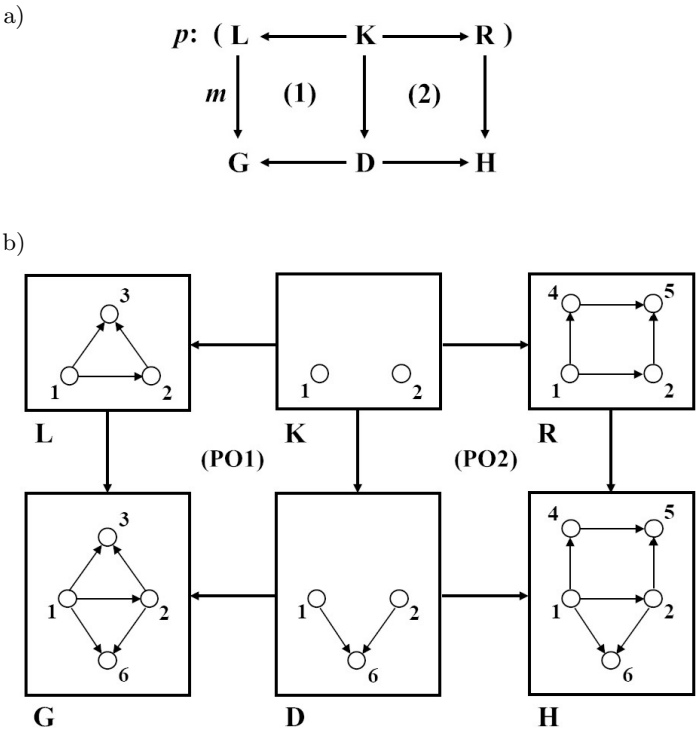


Figure 7. The scheme of DPO graph transformation (a) and its example (b)

Both an *indirect graph transformation* and the *graph language* defined by a DPO algebraic graph transformation system are defined in an analogous way as for graph grammars introduced above.

4. Applications of syntactic pattern recognition models in bioinformatics

The survey of the applications of syntactic pattern recognition methods in bioinformatics is presented in this section. Due to the presentation of string-based models, tree-based models and graph-based models in subsections: 3.1, 3.2 and 3.3 of the previous section, we can just refer to these models during the survey of their applications below in subsections: 4.1, 4.2 and 4.3, respectively. The summary of these applications is included in Subsection 4.4.

4.1. Applications of string-based models

In the area of bioinformatics, syntactic pattern recognition methods were applied firstly for chromosome analysis. The research team conducted by R.S. Ledley constructed the FIDAC system for scanning the chromosome photomicrographs for karyotype analysis in the 1960s [83,120,121]. The problem of biological images was studied by R.A. Kirsch [109]. K. S. Fu with collaborators led research into the analysis of photomicrographs of chromosomes in the 1970s. Precedence parsing [122] and stochastic context-free programmed grammars [71,92] were applied. In [192] the error-correcting recognition system was applied. The direct parsing model was presented in [189].

Research into the use of formal languages, grammars and automata in molecular biology and genetics was led in the 1980s and early 1990s [19,22,41,88,168,169]. tRNA modeling was performed with the help of *stochastic context-free grammars* (*CFGs*) [162]. For the parsing of DNA sequences, string variable grammars (an extension of definite clause grammars) were used in [170]. Syntactic pattern recognition-based methods were applied for the identification of regulatory sites in [159]. Stochastic *CFGs* were used for the modeling of RNA pseudoknot structures in [23]. Multiple sequence alignment was performed with the help of multi-tape S-attribute grammars in [123]. The inference of strictly locally testable languages for DNA sequence analysis was presented in [204].

In the area of gene expression and regulation, generative grammars have been used since 1989 [31–33]. Finite-state automata (and transducers) for applications in this area were presented in [22] and HMMs – in [208]. Context-sensitive grammars were applied for describing biological binding operators to model gene regulation in [13]. Modeling gene expression and regulation based on the operon model of Jacob and Monod with the help of finite-state automata was presented in [108]. Attributed context-free grammars were used for testing relationships between DNA sequences and phenotypes in [29]. The identification of the promoter regions with the help of context-free grammars was presented in [36].

Small subunit ribosomal RNA multiple alignments were constructed with the help of stochastic *CFGs* in [24]. The studies into the issue of predicting RNA secondary structures containing pseudoknots resulted in the proof of NP-completeness of this problem [133]. A polynomial time syntax analyzer for augmented *CFGs* generating

pseudoknotted structures was constructed in [157]. An ncRNA gene detection was made with the help of pair stochastic *CFGs* in [158]. Basic gene grammars were defined for processing DNA sequences in [124]. The model of grammatical induction for the recognition of human neuropeptide precursors was defined in [141]. A pairwise RNA structure comparison was made with stochastic *CFGs* in [89]. Parallel communicating grammars were constructed for modeling RNA pseudoknotted structures in [28]. The extraction of protein interaction information was made with the help of context-free grammars in [190]. An RNA secondary structure prediction with the help of stochastic *CFGs* was studied in [5,38,40,42,110,111]. Dependency grammars were used for the analysis of protein-protein interactions in [164]. Link grammars and they parsing were applied for the extraction of protein interaction information in [176]. The induction of even linear grammars was applied for predicting transmembrane domains in proteins in [147]. The prediction of RNA-RNA interaction was made with stochastic multiple *CFGs* in [105,107,175]. The studies of RNA pseudoknotted secondary structures with the help of multiple context-free grammars were presented in [49,143,155]. The analysis of protein sequences was performed with the help of stochastic *CFGs* in [45, 46]. The use of inference of regular grammars for larger-than-gene structures was discussed in [193]. Multi-dimensional (based on linear and context-free) grammars were used for DNA-protein alignment in [178]. A grammatical inference method was constructed for classification of amyloidogenic hexapeptides in [200].

Hidden Markov models (HMMs) are one of the most popular syntactic pattern recognition formal tools which are applied in bioinformatics [48,67]. HMMs were applied for gene/sequence prediction and modeling [26,27,103,117,126,140,142,154,188], sequence alignment [10, 144], protein secondary structure prediction [119, 202], base calling [128, 177], modeling sequencing errors [131], predicting transmembrane protein topology [116, 212], predicting and discriminating beta-barrel outer membrane proteins [6–8], RNA folding and alignment [86], ncRNA identification [180, 209], ncRNA annotation [21, 199], ncRNA structural alignment [206] and identification of protein domains [75, 191]. Novel models based on HMMs were defined in bioinformatics. The most popular ones include: *profile hidden Markov models* [3,15,47,87,90,98,104,115,148,150,183,184,186,187,201] which are used for representing and analyzing sequence profiles and *pair hidden Markov models* [39,44,112,144,198] which are applied for finding sequence alignments by emitting two (aligned) strings. *Generalized hidden Markov models* which emit a string at a state, were applied for gene prediction in [118, 134, 154]. Previously emitted substrings are used for determining the probabilities of future states in *context-sensitive hidden Markov models* [2, 206], which allows one to represent correlations between subsequences. (Standard) HMMs are combined with with continuous Markov chains to define *evolutionary hidden Markov models* [146] used to represent the evolution of biological sequences. Profile HMMs were applied for a viral discovery from metagenomic data in [4].

Fundamental studies into the use of syntactic pattern recognition in bioinformatics and comprehensive synthetics overview were presented in seminal monographs and papers. The most important include [171–173]. The applications of HMMs in

bioinformatics were summarized in [84,205] and the use of formal lingustics tools can be found in [9,35,44,149,165]. The problem of inferencing stochastic grammars from biological sequences was studied in [161]. The research into the applying of Natural Language Processing for genomics was presented in [203]. The studies into the possible existence of protein grammar which generates folding patterns in protein domains were presented in [151]. The application of computational linguistics for biopolymer structure studies was considered in [37].

4.2. Applications of tree-based models

Stochastic tree grammars were used for the prediction of protein secondary structures in [1, 135] and for the prediction of RNA/protein tertiary structures in [38]. Tree grammars were applied for the modeling of multiple biomolecular structures in [210, 211], for the computation of exact RNA shape probabilities in [93], for RNA analysis [129] and for RNA pseudoknot comparison in [152]. The mining of human-viral infection patterns was performed with the help of regular tree grammars in [182]. Rectangle tree grammars were used for predicting RNA secondary structures in [127].

RNA structure prediction with the help of *Tree Adjoining Grammars* (*TAGs*) was presented in [196]. TAGs were used for pseudoknot identification in [174]. The generating of RNA secondary structure including pseudoknots with the help of extended simple linear Tree Adjoining Grammars (ESL-TAG) was presented in [106]. This class of TAGs was used to construct an algorithm for tertiary interactions over pseudoknots for the predicting of RNA secondary structures in [91]. Pair stochastic Tree Adjoining Grammars (PSTAG) were used for a pseudoknot RNA structure prediction in [139]. The grammatical representation of macromolecular structures with the help of Tree Adjoining Grammars and related formalisms was proposed in [30].

Algebraic Dynamic Programming (ADP), based on tree grammars, was firstly used for RNA folding [76]. Its applications include: RNA folding [137], aligning recombinant DNA sequences [78], pairwise sequence comparison [77], RNA structure prediction and analysis [82] and the alignment of bio-structure tree representations [14, 80].

4.3. Applications of graph-based models

The induction (inference) of node label controlled (NLC) graph grammars was applied for analyzing protein sequence data in [99]. The modeling of protein structures with the help of neighborhood-controlled embedding (eNCE) graph grammars was presented in [197]. Algebraic (DPO) graph transformation systems were used for modeling RNA folding in [136]. These systems were also applied for analyzing metabolic networks in [179] and for modeling RNA tertiary structure motifs [185]. String-regulated rewriting graph grammars were used for genetic regulation in [130]. The use of an inference algorithm for k-testable graph languages in order to analyze hairpin RNA molecules data sets was presented in [73].

4.4. Summary of applications

A summary of applications of syntactic pattern recognition models in bioinformatics described in the previous subsections is presented in Table 1.

Table 1
The summary of applications of syntactic pattern recognition models in bioinformatics (in chronological order)

Model type	Models	References
String-based models	Regular grammars, stochastic context-free grammars, programmed context-free grammars, multiple context-free grammars, context-sensitive grammars, attributed grammars, finite-state automata, hidden Markov models, precedence parsing, CYK parsing, algebraic dynamic programming	[120], [121], [83], [109], [71], [122], [192], [22], [189], [88], [31–33], [168–173], [41], [115, 117], [162], [19], [23], [118], [13], [123], [159], [47, 48], [104], [204], [110], [24], [133], [154], [157, 158], [108], [116], [124], [141], [89], [144], [151], [203], [3], [28], [111], [112], [131], [146], [190], [6–8], [26,27], [42], [201], [39], [103], [134], [161], [183], [40], [105, 107], [142], [198], [148], [186,187], [199], [209], [15], [21], [37], [67], [86], [128], [164], [176], [184], [202], [147], [206], [29], [45], [180], [205], [2], [90], [98], [212], [193], [5], [143], [177], [191], [36], [46], [38], [75], [178], [4], [35], [155], [200], [119], [140], [188], [208], [149], [126], [49], [165]
Tree-based models	(Stochastic) regular tree grammars, Tree Adjoining Grammars, algebraic dynamic programming	[135], [1], [196], [76], [78], [77], [79], [81], [82], [106], [139], [30], [174], [93], [127], [210, 211], [91], [38], [80], [166], [182], [14], [129], [152], [137]
Graph-based models	*NLC* graph grammars, *NCE* graph grammars, algebraic (*DPO*) graph transformation systems, string-regulated rewriting graph grammars, k-testable graph languages	[99], [185], [130], [136], [73], [179], [197]

5. Conclusions

In Section 3 we have presented the basic formal tools of syntactic pattern recognition (SPR) which are used in bioinformatics. As one could see in Section 4, SPR models and methods, namely various classes of generative grammars, syntax analyzers of many types and a lot of language inference (induction) algorithms have been successfully used in bioinformatics. Indeed, the popularity of these methods resulting in plenty of applications in this research area is amazing. At the same time, bioinformatics is an interesting and challenging research area for computer scientists developing novel syntactic pattern recognition models for real-world applications.

References

[1] Abe N., Mamitsuka H.: Predicting protein secondary structure using stochastic tree grammars, *Machine Learning*, vol. 29, pp. 275–301, 1997.

[2] Agarwal S., Vaz C., Bhattacharya A., Srinivasan A.: Prediction of novel precursor miRNAs using a context-sensitive hidden Markov model (CSHMM), *BMC Bioinformatics*, vol. 11 (Suppl 1): S29, 2010. doi: 10.1186/1471-2105-11-s1-s29.

[3] Ahola V., Aittokallio T., Uusipaikka E., Vihinen M.: Efficient estimation of emission probabilities in profile hidden Markov models, *Bioinformatics*, vol. 19, pp. 2359–2368, 2003. doi: 10.1093/bioinformatics/btg328.

[4] Alves J.M.P., de Oliveira A.L., Sandberg T.O.M., Moreno-Gallego J.L., de Toledo M.A.F., de Moura E.M.M., Oliveira L.S., *et al.*: GenSeed-HMM: A tool for progressive assembly using profile HMMs as seeds and its application in alpavirinae viral discovery from metagenomic data, *Frontiers in Microbiology*, vol. 7, 269, 2016. doi: 10.3389/fmicb.2016.00269.

[5] Anderson J.W.J., Tataru P., Staines J., Hein J., Lyngsø R.: Evolving stochastic context-free grammars for RNA secondary structure prediction, *BMC Bioinformatics*, vol. 13, 78, 2012. doi: 10.1186/1471-2105-13-78.

[6] Bagos P.G., Liakopoulos T.D., Hamodrakas S.J.: Evaluation of methods for predicting the topology of β-barrel outer membrane proteins and a consensus prediction method, *BMC Bioinformatics*, vol. 6, 7, 2005. doi: 10.1186/1471-2105-6-7.

[7] Bagos P.G., Liakopoulos T.D., Spyropoulos I.C., Hamodrakas S.J.: A hidden Markov model method, capable of predicting and discriminating β-barrel outer membrane proteins, *BMC Bioinformatics*, vol. 5, 29, 2004. doi: 10.1186/1471-2105-5-29.

[8] Bagos P.G., Liakopoulos T.D., Spyropoulos I.C., Hamodrakas S.J.: PRED-TMBB: a web server for predicting the topology of β-barrel outer membrane proteins, *Nucleic Acids Research*, vol. 32, pp. W400–W404, 2004. doi: 10.1093/nar/gkh417.

[9] Baldi P., Brunak S.: *Bioinformatics: The Machine Learning Approach*, MIT Press, Cambridge, MA, 2001.

[10] Baldi P., Chauvin Y., Hunkapillar T., McClure M.: Hidden Markov models of biological primary sequence information, *Proceedings of the National Academy of Sciences of the USA*, vol. 91, pp. 1059–1063, 1994. doi: 10.1073/pnas.91.3.1059.

[11] Baum L.E., Petrie T.: Statistical inference for probabilistic functions of finite state Markov chains, *The Annals of Mathematical Statistics*, vol. 37, pp. 1554–1563, 1966. doi: 10.1214/aoms/1177699147.

[12] Bellman R.: *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957. doi: 10.2307/j.ctv1nxcw0f.

[13] Bentolila S.: A grammar describing 'biological binding operators' to model gene regulation, *Biochimie*, vol. 78, pp. 335–350, 1996. doi: 10.1016/0300-9084(96)84766-3.

[14] Berkemer S.J., zu Siederdissen Höner C., Stadler P.F.: Algebraic dynamic programming on trees, *Algorithms*, vol. 10, 135, 2017. doi: 10.3390/a10040135.

[15] Bernardes J.S., Dávila A.M.R., Costa V.S., Zaverucha G.: Improving model construction of profile HMMs for remote homology detection through structural alignment, *BMC Bioinformatics*, vol. 8, 435, 2007. doi: 10.1186/1471-2105-8-435.

[16] Bhargava B.K., Fu K.: Stochastic tree systems for syntactic pattern recognition. In: *Proceedings of Twelfth Annual Allerton Conference on Circuit and System Theory*, pp. 278–287, Monticello, IL, 1974.

[17] Booth T.L., Thompson R.A.: Applying probability measures to abstract languages, *IEEE Transactions on Computers*, vol. C-22(5), pp. 442–450, 1973. doi: 10.1109/t-c.1973.223746.

[18] Brainerd W.S.: Tree generating regular systems, *Information and Control*, vol. 14, pp. 217–231, 1969. doi: 10.1016/s0019-9958(69)90065-5.

[19] Bralley P.: An introduction to molecular linguistics, *BioScience*, vol. 46, pp. 146–153, 1996. doi: 10.2307/1312817.

[20] Brandenburg F.J.: On the complexity of the membership problem of graph grammars. In: *Proceedings of the Workshop on Graphtheoretic Concepts in Computer Science*, pp. 40–49, Osnabrück, Germany, 1983.

[21] Brejová B., Brown D.G., Vinař T.: The most probable annotation problem in HMMs and its application to bioinformatics, *Journal of Computer and System Sciences*, vol. 73, pp. 1060–1077, 2007. doi: 10.1016/j.jcss.2007.03.011.

[22] Brendel V., Busse H.G.: Genome structure described by formal languages, *Nucleic Acids Research*, vol. 12, pp. 2561–2568, 1984.

[23] Brown M., Wilson C.: RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. In: *Proceedings of 1996 Pacific Symposium on Biocomputing*, pp. 109–125, Hawaii, 1996.

[24] Brown M.P.: Small subunit ribosomal RNA modeling using stochastic context-free grammars. In: *Proceedings of 8th Internatonal Conference on Intelligent Systems for Molecular Biology*, pp. 57–66, San Diego, CA, USA, 2000.

[25] Bunke H., Sanfeliu A. (eds.): *Syntactic and Structural Pattern Recognition – Theory and Applications*, World Scientific, Singapore, 1990. doi: 10.1142/0580.

[26] Bystroff C., Krogh A.: Hidden Markov models for prediction of protein features. In: M. Zaki, C. Bystroff (eds.), *Protein Structure Prediction. Methods in Molecular Biology*, pp. 173–198, Humana Press, New Jersey, 2008. doi: 10.1007/978-1-59745-574-9_7.

[27] Bystroff C., Shao Y., Yuan X.: Five hierarchical levels of sequence-structure correlation in proteins, *Applied Bioinformatics*, vol. 3, pp. 97–104, 2004. doi: 10.2165/00822942-200403020-00004.

[28] Cai L., Malmberg R., Wu Y.: Stochastic modeling of RNA pseudoknotted structures: A grammatical approach, *Bioinformatics*, vol. 19, pp. i66–i73, 2003. doi: 10.1093/bioinformatics/btg1007.

[29] Cai Y., Lux M.W., Adam L., Peccoud J.: Modeling structure-function relationships in synthetic DNA sequences using attribute grammars, *PLoS Computational Biology*, vol. 5, e1000529, 2009. doi: 10.1371/journal.pcbi.1000529.

[30] Chiang D., Joshi A.K., Searls D.B.: Grammatical representations of macromolecular structure, *Journal of Computational Biology*, vol. 13, pp. 1077–1100, 2006. doi: 10.1089/cmb.2006.13.1077.

[31] Collado-Vides J.: A transformational-grammar approach to the study of the regulation of gene expression, *Journal of Theoretical Biology*, vol. 136, pp. 403–425, 1989. doi: 10.1016/s0022-5193(89)80156-0.

[32] Collado-Vides J.: A syntactic representation of units of genetic information – A syntax of units of genetic information, *Journal of Theoretical Biology*, vol. 148(3), pp. 401–429, 1991. doi: 10.1016/s0022-5193(05)80245-0.

[33] Collado-Vides J.: Grammatical model of the regulation of gene expression, *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89(20), pp. 9405–9409, 1992. doi: 10.1073/pnas.89.20.9405.

[34] Corn S.: Explicit definitions and linguistics dominoes. In: J.F. Hart, S. Takasu (eds.), *Systems and Computer Science*, pp. 77–115, University of Toronto Press, Toronto, 1967. doi: 10.3138/9781487592769-008.

[35] Coste F.: Learning the language of biological sequences. In: J. Heinz, J.M. Sempere (eds.), *Topics in Grammatical Inference*, pp. 215–247, Springer, 2016. doi: 10.1007/978-3-662-48395-4_8.

[36] Datta S., Mukhopadhyay S.: A composite method based on formal grammar and DNA structural features in detecting human polymerase II promoter region, *PLoS ONE*, vol. 8, e54843, 2013. doi: 10.1371/journal.pone.0054843.

[37] Dill K.E., Lucas A., Hockenmaier J., Huang L., Chiang D., Joshi A.K.: Computational linguistics: A new tool for exploring biopolymer structures and statistical mechanics, *Polymer*, vol. 48, pp. 4289–4300, 2007. doi: 10.1016/j.polymer.2007.05.018.

[38] Ding L., Samad A., Xue X., Huang X., Malmberg R.L., Cai L.: Stochastic k-tree grammar and its application in biomolecular structure modeling, *Lecture Notes in Computer Science*, vol. 8370, pp. 308–322, 2014. doi: 10.1007/978-3-319-04921-2_25.

[39] Do C.B., Mahabhashyam M.S., Brudno M., Batzoglou S.: ProbCons: Probabilistic consistency-based multiple sequence alignment, *Genome Research*, vol. 15, pp. 330–340, 2005. doi: 10.1101/gr.2821705.

[40] Do C.B., Woods D.A., Batzoglou S.: CONTRAfold: RNA secondary structure prediction without physics-based models, *Bioinformatics*, vol. 22, pp. e90–e98, 2006. doi: 10.1093/bioinformatics/btl246.

[41] Dong S., Searls D.B.: Gene structure prediction by linguistic methods, *Genomics*, vol. 23, pp. 540–551, 1994. doi: 10.1006/geno.1994.1541.

[42] Dowell R.D., Eddy S.R.: Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction, *BMC Bioinformatics*, vol. 5, 71, 2004. doi: 10.1186/1471-2105-5-71.

[43] Duda R.O., Hart P.E., Stork D.G.: *Pattern Classification*, 2nd ed., Wiley, New York, 2001.

[44] Durbin R., Eddy S.R., Krogh A., Mitchison G.: *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, Cambridge, UK, 2002.

[45] Dyrka W., Nebel J.C.: A stochastic context free grammar based framework for analysis of protein sequences, *BMC Bioinformatics*, vol. 10, 323, 2009. doi: 10.1186/1471-2105-10-323.

[46] Dyrka W., Nebel J.C., Kotulska M.: Probabilistic grammatical model of protein language and its application to helix-helix contact site classification, *Algorithms for Molecular Biology*, vol. 8, 31, 2013.

[47] Eddy S.R.: Profile hidden Markov models, *Bioinformatics*, vol. 14, pp. 755–763, 1998. doi: 10.1093/bioinformatics/14.9.755.

[48] Eddy S.R.: What is a hidden Markov model?, *Nature Biotechnology*, vol. 22, pp. 1315–1316, 2004. doi: 10.1038/nbt1004-1315.

[49] Eggers D., zu Siederdissen Höner H.C., Stadler P.F.: Accuracy of RNA structure prediction depends on the pseudoknot grammar. In: N.M. Scherer, R.C. de Melo-Minardi (eds.), *Advances in Bioinformatics and Computational Biology. BSB 2022*, Lecture Notes in Computer Science, vol. 13523, pp. 20–31, Springer, Cham, 2022. doi: 10.1007/978-3-031-21175-1_3.

[50] Ehrig H., Ehrig K., Prange U., Taentzer G.: *Fundamentals of Algebraic Graph Transformation*, Springer, Berlin-Heidelberg, 2006.

[51] Ehrig H., Engels G., Kreowski H.J., Rozenberg G. (eds.): *Handbook of Graph Grammars and Computing by Graph Transformation, Vol. 2: Applications, Languages and Tools*, World Scientific, Singapore, 1999. doi: 10.1142/9789812815149.

[52] Ehrig H., Kreowski H.J.: Pushout-Properties: An analysis of gluing constructions for graphs, *Mathematische Nachrichten*, vol. 91(1), pp. 135–149, 1979. doi: 10.1002/mana.19790910111.

[53] Ehrig H., Pfender M., Schneider H.J.: Graph grammars: An algebraic approach. In: *Proceedings of 14th Annual IEEE Symposium on Switching and Automata Theory*, pp. 167–180, 1973. doi: 10.1109/swat.1973.11.

[54] Flasiński M.: Parsing of edNLC-graph grammars for scene analysis, *Pattern Recognition*, vol. 21, pp. 623–629, 1988. doi: 10.1016/0031-3203(88)90034-9.

[55] Flasiński M.: Distorted pattern analysis with the help of Node Label Controlled graph languages, *Pattern Recognition*, vol. 23, pp. 765–774, 1990. doi: 10.1016/0031-3203(90)90099-7.

[56] Flasiński M.: On the parsing of deterministic graph languages for syntactic pattern recognition, *Pattern Recognition*, vol. 26, pp. 1–16, 1993. doi: 10.1016/0031-3203(93)90083-9.

[57] Flasiński M.: Use of graph grammars for the description of mechanical parts, *Computer-Aided Design*, vol. 27, pp. 403–433, 1995. doi: 10.1016/0010-4485(94)00015-6.

[58] Flasiński M.: Power properties of NLC graph grammars with a polynomial membership problem, *Theoretical Computer Science*, vol. 201, pp. 189–231, 1998. doi: 10.1016/s0304-3975(97)00212-0.

[59] Flasiński M.: Inference of parsable graph grammars for syntactic pattern recognition, *Fundamenta Informaticae*, vol. 80, pp. 379–413, 2007.

[60] Flasiński M.: *Introduction to Artificial Intelligence*, Springer International, Switzerland, 2016. doi: 10.1007/978-3-319-40022-8.

[61] Flasiński M.: *Syntactic Pattern Recognition*, World Scientific, New Jersey-London-Singapore, 2019.

[62] Flasiński M., Jurek J.: Dynamically programmed automata for quasi context sensitive languages as a tool for inference support in pattern recognition-based real-time control expert systems, *Pattern Recognition*, vol. 32, pp. 671–690, 1999. doi: 10.1016/s0031-3203(98)00115-0.

[63] Flasiński M., Kotulski L.: On the use of graph grammars for the control of a distributed software allocation, *The Computer Journal*, vol. 35, pp. A165–A175 1992.

[64] Flasiński M., Lewicki G.: The convergent method of constructing polynomial discriminant functions for pattern recognition, *Pattern Recognition*, vol. 24, pp. 1009–1015, 1991. doi: 10.1016/0031-3203(91)90098-p.

[65] Flasiński M., Myśliński S.: On the use of graph parsing for recognition of isolated hand postures of Polish Sign Language, *Pattern Recognition*, vol. 43, pp. 2249–2264, 2010. doi: 10.1016/j.patcog.2010.01.004.

[66] Flasiński M., Jurek J., Peszek T.: Multi-derivational parsing of vague languages – the new paradigm of syntactic pattern recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, 2024. doi: 10.1109/tpami.2024.3367245.

[67] Fonzo de V., Aluffi-Pentini F., Parisi V.: Hidden Markov models in bioinformatics, *Current Bioinformatics*, vol. 2(1), pp. 49–61, 2007. doi: 10.2174/1574893 07779314348.

[68] Fu K.S.: Stochastic automata, stochastic languages and pattern recognition, *Journal of Cybernetics*, vol. 1, pp. 31–49, 1971.

[69] Fu K.S.: Stochastic tree languages and their application to picture processing. In: P.R. Krishnaiah (ed.), *Multivariate Analysis V*, pp. 561–579, North-Holland, Amsterdam, 1980.

[70] Fu K.S.: *Syntactic Pattern Recognition and Applications*, Prentice Hall, Englewood Cliffs, 1982.

[71] Fu K.S., Huang T.: Stochastic grammars and languages, *International Journal of Computer and Information Sciences*, vol. 1, pp. 135–170, 1972. doi: 10.1007/bf00995736.

[72] Fu K.S., Li T.: On stochastic automata and languages, *Information Sciences*, vol. 1, pp. 403–419, 1969. doi: 10.1016/0020-0255(69)90024-3.

[73] Gallego A.J., López D., Calera-Rubio J.: Grammatical inference of directed acyclic graph languages with polynomial time complexity, *Journal of Computer and System Sciences*, vol. 95, pp. 19–34, 2018. doi: 10.1016/j.jcss.2017.12.002.

[74] Gécseg F., Steinby M.: *Tree Automata*, Akadémiai Kiadó, Budapest, 1984.

[75] Ghouila A., Florent I., Guerfali F.Z., Terrapon N., Laouini D., Yahia S.B., Gascuel O., *et al.*: Identification of Divergent Protein Domains by Combining HMM-HMM Comparisons and Co-Occurrence Detection, *PLoS ONE*, vol. 9, e95275, 2014. doi: 10.1371/journal.pone.0095275.

[76] Giegerich R.: *A declarative approach to the development of dynamic programming algorithms, applied to RNA folding*, Tech. rep., Bielefeld University, Germany, 1998.

[77] Giegerich R.: Explaining and controlling ambiguity in dynamic programming, *Lecture Notes in Computer Science*, vol. 1848, pp. 46–59, 2000. doi: 10.1007/3-540-45123-4_6.

[78] Giegerich R.: A systematic approach to dynamic programming in bioinformatics, *Bioinformatics*, vol. 16(8), pp. 665–677, 2000. doi: 10.1093/bioinformatics/16.8.665.

[79] Giegerich R., Meyer C.: Algebraic Dynamic Programming, *Lecture Notes in Computer Science*, vol. 2422, pp. 349–364, 2002. doi: 10.1007/3-540-45719-4_24.

[80] Giegerich R., Touzet H.: Modeling dynamic programming problems over sequences and trees with inverse coupled rewrite systems, *Algorithms*, vol. 7, pp. 62–144, 2014. doi: 10.3390/a7010062.

[81] Giegerich R., Meyer C., Steffen P.: Towards a discipline of dynamic programming, *Lecture Notes in Informatics*, vol. P-147, pp. 3–44, 2002.

[82] Giegerich R., Meyer C., Steffen P.: A discipline of dynamic programming over sequence data, *Science of Computer Programming*, vol. 51, pp. 215–263, 2004. doi: 10.1016/j.scico.2003.12.005.

[83] Golab T., Ledley R.S., Rotolo L.S.: FIDAC: Film input to digital automatic computer, *Pattern Recognition*, vol. 3, pp. 123–156, 1971. doi: 10.1016/0031-3203(71)90035-5.

[84] Gollery M. (ed.): *Handbook of Hidden Markov Models in Bioinformatics*, Chapman and Hall / CRC, Boca Raton, FL, 2008. doi: 10.1201/9781420011807.

[85] Grenander U.: *Syntax-controlled probabilities*, Tech. rep., Brown University, Providence, R.I., 1967.

[86] Harmanci A.O., Sharma G., Mathews D.H.: Efficient pairwise RNA structure prediction using probabilistic alignment constraints in *Dynalign*, *BMC Bioinformatics*, vol. 8, 130, 2007.

[87] Haussler D., Krogh A., Mian I.S., Sjöander K.: Protein modeling using hidden Markov models: analysis of globins. In: *Proceedings of 26th Annual Hawaii International Conference on Systems Sciences*, pp. 792–802, 1993.

[88] Head T.: Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors, *Bulletin of Mathematical Biology*, vol. 49, pp. 737–759, 1987. doi: 10.1016/s0092-8240(87)90018-8.

[89] Holmes I., Rubin G.M.: Pairwise RNA structure comparison with stochastic context-free grammars. In: *Proceedings of 2002 Pacific Symposium on Biocomputing*, pp. 163–174, Hawaii, 2002.

[90] Horan K., Shelton C., Girke T.: Predicting conserved protein motifs with SubHMMs, *BMC Bioinformatics*, vol. 11, 205, 2010. doi: 10.1186/1471-2105-11-205.

[91] Hsu B.Y., Wong T.K.F., Hon W.K., Liu X., Lam T.W., Yiu S.M.: A Local Structural Prediction Algorithm for RNA Triple Helix Structure. In: A. Ngom, E. Formenti, J.K. Hao, X.M. Zhao, T. van Laarhoven (eds.), *Pattern Recognition in Bioinformatics. PRIB 2013.* Lecture Notes in Computer Science, vol. 7968, pp. 102–113, Springer, Berlin–Heidelberg, 2013. doi: 10.1007/978-3-642-39159-0_10.

[92] Huang T., Fu K.S.: Stochastic syntactic analysis for programmed grammars and syntactic pattern recognition, *Computer Graphics and Image Processing*, vol. 1, pp. 257–283, 1972. doi: 10.1016/s0146-664x(72)80018-2.

[93] Janssen S., Giegerich R.: Faster computation of exact RNA shape probabilities, *Bioinformatics*, vol. 26, pp. 632–639, 2010. doi: 10.1093/bioinformatics/btq014.

[94] Janssens D., Rozenberg G.: On the structure of node-label-controlled graph languages, *Information Sciences*, vol. 20, pp. 191–216, 1980. doi: 10.1016/0020-0255(80)90038-9.

[95] Janssens D., Rozenberg G.: Restrictions, extensions, and variations of NLC grammars, *Information Sciences*, vol. 20, pp. 217–244, 1980. doi: 10.1016/0020-0255(80)90039-0.

[96] Janssens D., Rozenberg G.: Graph grammars with neighbourhood-controlled embedding, *Theoretical Computer Science*, vol. 21, pp. 55–74, 1982. doi: 10.1016/0304-3975(82)90088-3.

[97] Janssens D., Rozenberg G., Verraedt R.: On sequential and parallel node-rewriting graph grammars, *Computer Graphics and Image Processing*, vol. 18, pp. 279–304, 1982. doi: 10.1016/0146-664x(82)90036-3.

[98] Johnson L.S., Eddy S.R., Portugaly L.: Hidden Markov model speed heuristic and iterative HMM search procedure, *BMC Bioinformatics*, vol. 11, 431, 2010. doi: 10.1186/1471-2105-11-431.

[99] Jonyer I., Holder L.B., Cook D.J.: MDL-based context-free graph grammar induction and applications, *International Journal on Artificial Intelligence Tools*, vol. 13, pp. 65–79, 2004. doi: 10.1142/s0218213004001429.

[100] Joshi A.K.: Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In: D.R. Dowty, L. Karttunen, A.M. Zwicky (eds.), *Natural Language Processing: Theoretical, Computational and Psychological Perspective*, pp. 206–250, Cambridge University Press, New York, NY, 1985.

[101] Joshi A.K., Levy L.S., Takahashi M.: Tree adjunct grammars, *Journal of Computer and System Sciences*, vol. 10, pp. 136–163, 1975. doi: 10.1016/s0022-0000(75)80019-5.

[102] Joshi A.K., Schabes Y.: Tree adjoining grammars. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages – III*, pp. 69–123, Springer, New York, NY, 1997. doi: 10.1007/978-3-642-59126-6_2.

[103] Käll L., Krogh A., Sonnhammer E.: An HMM posterior decoder for sequence feature prediction that includes homology information, *Bioinformatics*, vol. 21, pp. i251–i257, 2005. doi: 10.1093/bioinformatics/bti1014.

[104] Karplus K., Barrett C., Hughey R.: Hidden Markov models for detecting remote protein homologies, *Bioinformatics*, vol. 14, pp. 846–856, 1998. doi: 10.1093/bioinformatics/14.10.846.

[105] Kato Y., Akutsu T., Seki H.: A grammatical approach to RNA-RNA interaction prediction, *Pattern Recognition*, vol. 42, pp. 531–538, 2009. doi: 10.1016/j.patcog.2008.08.004.

[106] Kato Y., Seki H., Kasami T.: Subclasses of tree adjoining grammars for RNA secondary structure. In: *Proceedings of 7th International Workshop on Tree Adjoining Grammar and Related Formalisms*, pp. 48–55, Vancouver, Canada, 2004.

[107] Kato Y., Seki H., Kasami T.: Stochastic multiple context-free grammar for RNA pseudoknot modeling. In: *Proceedings of 8th International Workshop on Tree Adjoining Grammar and Related Formalisms*, pp. 57–64, Sydney, Australia, 2006. doi: 10.3115/1654690.1654698.

[108] Kennedy P.J., Osborn T.R.: A model of gene expression and regulation in an artificial cellular organism, *Complex Systems*, vol. 13, pp. 33–59, 2001.

[109] Kirsch R.A.: Computer determination of the constituent structure of bio1ogical images, *Computers and Biomedical Research*, vol. 4, pp. 315–328, 1971. doi: 10.1016/0010-4809(71)90034-6.

[110] Knudsen B., Hein J.: RNA secondary structure prediction using stochastic context-free grammars and evolutionary history, *Bioinformatics*, vol. 15, pp. 446–454, 1999. doi: 10.1093/bioinformatics/15.6.446.

[111] Knudsen B., Hein J.: Pfold: RNA secondary structure prediction using stochastic context-free grammars, *Nucleic Acids Research*, vol. 31, pp. 3423–3428, 2003.

[112] Knudsen B., Miyamoto M.M.: Sequence alignments and pair hidden Markov models using evolutionary history, *Journal of Molecular Biology*, vol. 333, pp. 453–460, 2003. doi: 10.1016/j.jmb.2003.08.015.

[113] Knuth D.: Semantics of context-free languages, *Mathematical Systems Theory*, vol. 2, pp. 127–145, 1968. doi: 10.1007/bf01692511.

[114] Koutroumbas K., Theodoridis S.: *Pattern Recognition*, 4th ed., Academic Press, Boston, 2008.

[115] Krogh A., Brown M., Mian I.S., Sjöander K., Haussler D.: Hidden Markov models in computational biology: Applications to protein modeling, *Journal of Molecular Biology*, vol. 235, pp. 1501–1531, 1994.

[116] Krogh A., Larsson B., Heijne von G., Sonnhammer E.: Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes, *Journal of Molecular Biology*, vol. 305, pp. 567–580, 2001.

[117] Krogh A., Mian I.S., Haussler D.: A hidden Markov model that finds genes in E. coli DNA, *Nucleic Acids Research*, vol. 22, pp. 4768–4778, 1994.

[118] Kulp D., Haussler D., Reese M.G., Eeckman F.H.: A Generalized Hidden Markov Model for the Recognition of Human Genes in DNA. In: *Proceedings of the 4th International Conference on Intelligent Systems for Molecular Biology*, pp. 134–142, St. Louis, MO, USA, 1996.

[119] Lasfar M., Bouden H.: A method of data mining using Hidden Markov Models (HMMs) for protein secondary structure prediction, *Procedia Computer Science*, vol. 127, pp. 42–51, 2018. doi: 10.1016/j.procs.2018.01.096.

[120] Ledley R.S.: High-speed automatic analysis of biomedical pictures, *Science,* vol 146, pp. 216–223, 1964. doi: 10.1126/science.146.3641.216.

[121] Ledley R.S., Rotolo L.S., Golab T.J., Jacobsen J.D., Ginsberg M.D., Wilson J.B.: FIDAC: Film input to digital automatic computer and associated syntax-directed pattern-recognition programming system. In: J.T. Tippet, D. Beckovitz, L. Clapp, C. Koester, A. Vanderburgh Jr. (eds.), *Optical and Electro-optical Information Processing*, pp. 591–613, MIT Press, Cambridge, MA, 1965.

[122] Lee H.C., Fu K.S.: A stochastic syntax analysis procedure and its application to pattern classification, *IEEE Transactions on Computers*, vol. 21, pp. 660–666, 1972. doi: 10.1109/t-c.1972.223571.

[123] Lefebvre F.: A grammar-based unification of several alignment and folding algorithms. In: *Proceedings of 4th International Conference on Intelligent Systems for Molecular Biology*, pp. 143–154, St. Louis, MO, USA, 1996.

[124] Leung S., Mellish C., Robertson D.: Basic Gene Grammars and DNA-Chart-Parser for language processing of *Escherichia coli* promoter DNA sequences, *Bioinformatics*, vol. 17, pp. 226–236, 2001.

[125] Levenshtein V.I.: Binary codes capable of correcting deletions, insertions and reversals, *Soviet Physics Doklady*, vol. 10, pp. 707–710, 1966.

[126] Li J., Lee J., Liao L.: A new algorithm to train hidden Markov models for biological sequences with partial labels, *BMC Bioinformatics*, vol. 22, 162, 2021. doi: 10.1186/s12859-021-04080-0.

[127] Li M., Cheng M., Ye Y., Hon W., Ting H., Lam T., Tang C., *et al.*: Predicting RNA secondary structures: One-grammar-fits-all solution, *Lecture Notes in Computer Science*, vol. 9096, pp. 211–222, 2015. doi: 10.1007/978-3-319-19048-8_18.

[128] Liang K.C., Wang X., Anastassiou D.: Bayesian Basecalling for DNA Sequence Analysis Using Hidden Markov Models, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4(3), pp. 430–440, 2007. doi: 10.1109/tcbb.2007.1027.

[129] Liu L., Mori T., Zhao Y., Hayashida M., Akutsu T.: Euler string-based compression of tree-structured data and its application to analysis of RNAs, *Current Bioinformatics*, vol. 13, pp. 25–33, 2018. doi: 10.2174/1574893611666160608102231.

[130] Lobo D., Vico F.J., Dassow J.: Graph grammars with string-regulated rewriting, *Theoretical Computer Science*, vol. 412, pp. 6101–6111, 2011. doi: 10.1016/j.tcs.2011.07.004.

[131] Lottaz C., Iseli C., Jongeneel C.V., Bucher P.: Modeling sequencing errors by combining hidden Markov models, *Bioinformatics*, vol. 19 (Suppl. 2), pp. i103–i112, 2003. doi: 10.1093/bioinformatics/btg1067.

[132] Lu S.Y., Fu K.S.: Structure-preserved error-correcting tree automata for syntactic pattern recognition. In: *1976 IEEE Conference on Decision and Control including the 15th Symposium on Adaptive Processes*, pp. 413–419, Clearwater, FL, USA, 1976. doi: 10.1109/cdc.1976.267768.

[133] Lyngsø R.B., Pedersen C.N.: RNA pseudoknot prediction in energy-based models, *Journal of Computational Biology*, vol. 7, pp. 409–427, 2000. doi: 10.1089/106652700750050862.

[134] Majoros W.H., Pertea M., Delcher A.L., Salzberg S.L.: Efficient decoding algorithms for generalized hidden Markov model gene finders, *BMC Bioinformatics*, vol. 6, 16, 2005. doi: 10.1186/1471-2105-6-16.

[135] Mamitsuka H., Abe N.: Predicting location and structure of beta-sheet regions using stochastic tree grammars. In: *Proceedings of 2nd International Conference on Intelligent Systems for Molecular Biology*, pp. 276–284, Stanford, CA, USA, 1994.

[136] Mamuye A., Merelli E., Tesei L.: A graph grammar for modelling RNA folding. In: *Proceedings of 2nd Graphs as Models Workshop*, pp. 31–41, Eindhoven, The Netherlands, 2016. doi: 10.4204/eptcs.231.3.

[137] Marchand B., Will S., Berkemer S.J., Ponty Y., Bulteau L.: Automated design of dynamic programming schemes for RNA folding with pseudoknots. In: *Proceedings of 22nd International Workshop on Algorithms in Bioinformatics*, pp. 7:1–7:24, Potsdam, Germany, 2022. doi: 10.1186/s13015-023-00229-z.

[138] Markov A.A.: Essai d'une recherche statistique sur le texte du roman *Eugene Onegin* illustrant la liaison des epreuve en chain, *Bulletin de l'Académie Impériale des Sciences de St-Pétersbourg*, vol. 7, pp. 153–162, 1913.

[139] Matsui H., Sato K., Sakakibara Y.: Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures, *Bioinformatics*, vol. 21, pp. 2611–2617, 2005. doi: 10.1093/bioinformatics/bti385.

[140] Menichelli C., Gascuel O., Bréhélin L.: Improving pairwise comparison of protein sequences with domain co-occurrence, *PLoS Computational Biology*, vol. 14, e1005889, 2018. doi: 10.1371/journal.pcbi.1005889.

[141] Muggleton S., Bryant C., Srinivasan A., Whittaker A., Topp S., Rawlings C.: Are grammatical representations useful for learning from biological sequence data? A case study, *Journal of Computational Biology*, vol. 8, pp. 493–522, 2001.

[142] Munch K., Krogh A.: Automatic generation of gene finders for eukaryotic species, *BMC Bioinformatics*, vol. 7, 263, 2006. doi: 10.1186/1471-2105-7-263.

[143] Nebel M.E., Weinberg F.: Algebraic and combinatorial properties of common RNA pseudoknot classes with applications, *Journal of Computational Biology*, vol. 19, pp. 1134–1150, 2012. doi: 10.1089/cmb.2011.0094.

[144] Pachter L., Alexandersson M., Cawley S.: Applications of generalized pair hidden Markov models to alignment and gene finding problems, *Journal of Computational Biology*, vol. 9, pp. 389–399, 2002. doi: 10.1089/10665270252935520.

[145] Pavlidis T.: Structural descriptions and graph grammars. In: S.K. Chang, K.S. Fu (eds.), *Pictorial Information Systems*, pp. 86–103, Springer, Berlin – Heidelberg – New York, 1980. doi: 10.1007/3-540-09757-0_4.

[146] Pedersen J.C., Hein J.: Gene finding with a hidden Markov model of genome structure and evolution, *Bioinformatics*, vol. 19, pp. 219–227, 2003. doi: 10.1093/bioinformatics/19.2.219.

[147] Peris P., López D., Campos M.: IgTM: An algorithm to predict transmembrane domains and topology in proteins, *BMC Bioinformatics*, vol. 9, 367, 2008. doi: 10.1186/1471-2105-9-367.

[148] Plötz T., Fink G.A.: Pattern recognition methods for advanced stochastic protein sequence analysis using HMMs, *Pattern Recognition*, vol. 39, pp. 2267–2280, 2006. doi: 10.1016/j.patcog.2005.10.007.

[149] Ponty Y.: *Ensemble Algorithms and Analytic Combinatorics in RNA Bioinformatics and Beyond*, Université Paris-Saclay, 2020.

[150] Porter T., Hajibabaei M.: Profile hidden Markov model sequence analysis can help remove putative pseudogenes from DNA barcoding and metabarcoding datasets, *BMC Bioinformatics*, vol. 22, 256, 2021. doi: 10.1186/s12859-021-04180-x.

[151] Przytycka T., Srinivasan R., Rose G.D.: Recursive domains in proteins, *Protein Science*, vol. 11, pp. 409–417, 2002. doi: 10.1110/ps.24701.

[152] Quadrini M., Tesei L., Merelli E.: An algebraic language for RNA pseudoknots comparison, *BMC Bioinformatics*, vol. 20, 161, 2019. doi: 10.1186/s12859-019-2689-5.

[153] Rabin M.O.: Probabilistic automata, *Information and Control*, vol. 6, pp. 230–245, 1963. doi: 10.1016/s0019-9958(63)90290-0.

[154] Reese M.G., Kulp D., Tammana H., Haussler D.: *Genie* – gene finding in *Drosophila melanogaster*, *Genome Research*, vol. 10, pp. 529–538, 2000.

[155] Riechert M., zu Siederdissen Höner C.H., Stadler P.F.: Algebraic dynamic programming for multiple context-free grammars, *Theoretical Computer Science*, vol. 639, pp. 91–109, 2016. doi: 10.1016/j.tcs.2016.05.032.

[156] Ripley B.D.: *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 2008.

[157] Rivas E., Eddy S.R.: The language of RNA: a formal grammar that includes pseudoknots, *Bioinformatics*, vol. 16(4), pp. 334–340, 2000. doi: 10.1093/bioinformatics/16.4.334.

[158] Rivas E., Eddy S.R.: Noncoding RNA gene detection using comparative sequence analysis, *BMC Bioinformatics*, vol. 2, 8, 2001. doi: 10.1186/1471-2105-2-8.

[159] Rosenblueth D.A., Thieffry D., Huerta A.M., Salgado H., Collado-Vides J.: Syntactic recognition of regulatory regions in *Escherichia coli*, *Computer Applications in the Biosciences*, vol. 12, pp. 415–422, 1996.

[160] Rosenkrantz D.J.: Programmed grammars and classes of formal languages, *Journal of the Association for Computing Machinery*, vol. 16, pp. 107–131, 1969. doi: 10.1145/321495.321504.

[161] Sakakibara Y.: Grammatical inference in bioinformatics, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1051–1062, 2005. doi: 10.1109/tpami.2005.140.

[162] Sakakibara Y., Brown M., Hughey R., Mian I.S., Sjölander K., Underwood R.C., Haussler D.: Stochastic context-free grammars for tRNA modeling, *Nuclear Acids Research*, vol. 22(23), pp. 5112–5120, 1994. doi: 10.1093/nar/22.23.5112.

[163] Salomaa A.: Probabilistic and weighted grammars, *Information and Control*, vol. 15, pp. 529–544, 1969. doi: 10.1016/s0019-9958(69)90554-3.

[164] Sanchez-Graillet O., Poesio M.: Negation of protein-protein interactions: analysis and extraction, *Bioinformatics*, vol. 23, pp. i424–i432, 2007. doi: 10.1093/bioinformatics/btm184.

[165] Sato K., Hamada M.: Recent trends in RNA informatics: a review of machine learning and deep learning for RNA secondary structure prediction and RNA drug discovery, *Briefings in Bioinformatics*, vol. 24, 2023. doi: 10.1093/bib/bbad186.

[166] Sauthoff G., Giegerich R.: Yield grammar analysis and product optimization in a domain-specific language for dynamic programming, *Science of Computer Programming*, vol. 87, pp. 2–22, 2014. doi: 10.1016/j.scico.2013.09.011.

[167] Schalkoff R.: *Pattern Recognition: Statistical, Structural and Neural Approaches*, Wiley, New York, 2005.

[168] Searls D.B.: The linguistics of DNA, *American Scientist*, vol. 80(6), pp. 579–591, 1992.

[169] Searls D.B.: The computational linguistics of biological sequences. In: L. Hunter (ed.), *Artificial Intelligence and Molecular Biology*, pp. 47–120, AAAI/MIT Press, Menlo Park, CA, 1993.

[170] Searls D.B.: String Variable Grammar: a logic grammar formalism for DNA sequences, *The Journal of Logic Programming*, vol. 24, pp. 73–102, 1995.

[171] Searls D.B.: Linguistic approaches to biological sequences, *Bioinformatics*, vol. 13, pp. 333–344, 1997. doi: 10.1093/bioinformatics/13.4.333.

[172] Searls D.B.: Reading the book of life, *Bioinformatics*, vol. 17, pp. 579–580, 2001. doi: 10.1093/bioinformatics/17.7.579.

[173] Searls D.B.: The language of genes, *Nature*, vol. 420, pp. 211–217, 2002. doi: 10.1038/nature01255.

[174] Seesi S.A., Rajasekaran S., Ammar R.: Pseudoknot Identification through Learning TAG_{RNA}. In: M. Chetty, A. Ngom, S. Ahmad (eds.), *Pattern Recognition in Bioinformatics. PRIB 2008.* Lecture Notes in Computer Science, vol. 5265, pp. 132–143, Springer, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-88436-1_12.

[175] Seki H., Matsumura T., Fujii M., Kasami T.: On multiple context-free grammars, *Theoretical Computer Science*, vol. 88, pp. 191–229, 1991. doi: 10.1016/0304-3975(91)90374-b.

[176] Seoud R.A.A., Youssef A.B.M., Kadah Y.M.: Extraction of protein interaction information from unstructured text using a link grammar parser. In: *Proceedings of 2007 International Conference on Computer Engineering and Systems*, pp. 70–75, Cairo, Egypt, 2007. doi: 10.1109/icces.2007.4447028.

[177] Shen X., Vikalo H.: ParticleCall: A particle filter for base calling in next-generation sequencing systems, *BMC Bioinformatics*, vol. 13, 160, 2012. doi: 10.1186/1471-2105-13-160.

[178] Siederdissen zu C.H., Hofacker I.L., Stadler P.F.: Product grammars for alignment and folding, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, pp. 507–519, 2015. doi: 10.1109/tcbb.2014.2326155.

[179] Silva da W.M.C., Andersen J.L., Holanda M.T., Walter M.E.M.T., Brigido M.M., Stadler P.F., Flamm C.: Exploring plant sesquiterpene diversity by generating chemical networks, *Processes*, vol. 7(4), 240, 2019. doi: 10.3390/pr7040240.

[180] Singh P., Bandyopadhyay P., Bhattacharya S., Krishnamachari A., Sengupta S.: Riboswitch detection using profile hidden Markov models, *BMC Bioinformatics*, vol. 10, 325, 2009. doi: 10.1186/1471-2105-10-325.

[181] Slisenko A.O.: Context-free grammars as a tool for describing polynomial-time subclasses of hard problems, *Information Processing Letters*, vol. 14, pp. 52–56, 1982. doi: 10.1016/0020-0190(82)90086-2.

[182] Smoly I., Carmel A., Shemer-Avni Y., Yeger-Lotem E., Ziv-Ukelson M.: Algorithms for regular tree grammar network search and their application to mining human-viral infection patterns, *Journal of Computational Biology*, vol. 23, pp. 165–179, 2016. doi: 10.1089/cmb.2015.0168.

[183] Söding J.: Protein homology detection by HMM-HMM comparison, *Bioinformatics*, vol. 21, pp. 951–960, 2005.

[184] Srivastava P.K., Desai D.K., Nandi S., Lynn A.M.: HMM-ModE-improved classification using profile hidden Markov models by optimising the discrimination threshold and modifying emission probabilities with negative training sequences, *BMC Bioinformatics*, vol. 8, 104, 2007. doi: 10.1186/1471-2105-8-104.

[185] St-Onge K., Thibault P., Hamel S., Major F.: Modeling RNA tertiary structure motifs by graph-grammars, *Nucleic Adids Research*, vol. 35, pp. 1726–1736, 2007. doi: 10.1093/nar/gkm069.

[186] Sun Y., Buhler J.: Designing patterns for profile HMM search, *Bioinformatics*, vol. 23, pp. e36–e43, 2006. doi: 10.1093/bioinformatics/btl323.

[187] Sun Y., Buhler J.: Designing patterns and profiles for faster HMM search, *IEEE/ACM Trans Computational Biology and Bioinformatics*, vol. 6, pp. 232–243, 2009. doi: 10.1109/tcbb.2008.14.

[188] Tamposis I.A., Tsirigos K.D., Theodoropoulou M.C., Kontou P.I., Bagos P.G.: Semi-supervised learning of hidden Markov models for biological sequence analysis, *Bioinformatics*, vol. 35, pp. 2208–2215, 2019. doi: 10.1093/bioinformatics/bty910.

[189] Tanaka E., Ikeda M., Ezure K.: Direct parsing, *Pattern Recognition*, vol. 19, pp. 315–323, 1986. doi: 10.1016/0031-3203(86)90057-9.

[190] Temkin J.M., Gilder M.R.: Extraction of protein interaction information from unstructured text using a context-free grammar, *Bioinformatics*, vol. 19, pp. 2046–2053, 2003. doi: 10.1093/bioinformatics/btg279.

[191] Terrapon N., Gascuel O., Maréchal, É., Bréhélin L.: Fitting hidden Markov models of protein domains to a target species: application to *Plasmodium falciparum*, *BMC Bioinformatics*, vol. 13, 67, 2012. doi: 10.1186/1471-2105-13-67.

[192] Thomason M.G., Gonzales R.C.: Syntactic recognition of imperfectly specified patterns, *IEEE Transactions on Computers*, vol. 24, pp. 93–95, 1975. doi: 10.1109/t-c.1975.224086.

[193] Tsafnat G., Schaeffer J., Clayphan A., Iredell J.R., Partridge S.R., Coiera E.: Computational inference of grammars for larger-than-gene structures from annotated gene sequences, *Bioinformatics*, vol. 27, pp. 791–796, 2011. doi: 10.1093/bioinformatics/btr036.

[194] Turakainen P.: On stochastic languages, *Information and Control*, vol. 12, pp. 304–313, 1968. doi: 10.1016/s0019-9958(68)90360-4.

[195] Turán G.: On the complexity of graph grammars, *Acta Cybernetica*, vol. 6(3), pp. 271–280, 1983.

[196] Uemura Y., Hasegawa A., Kobayashi S., Yokomori T.: Tree adjoining grammars for RNA structure prediction, *Theoretical Computer Science*, vol. 210, pp. 277–303, 1999. doi: 10.1016/s0304-3975(98)00090-5.

[197] Vijayakumar J., Mathew L., Nagar A.K.: A new class of graph grammars and modelling of certain biological structures, *Symmetry*, vol. 15, p. 349, 2023. doi: 10.3390/sym15020349.

[198] Wang J., Keightley P.D., Johnson T.: MCALIGN2: faster, accurate global pairwise alignment of non-coding DNA sequences based on explicit models of indel evolution, *BMC Bioinformatics*, vol. 7, 292, 2006. doi: 10.1186/1471-2105-7-292.

[199] Weinberg Z., Ruzzo W.L.: Sequence-based heuristics for faster annotation of non-coding RNA families, *Bioinformatics*, vol. 22, pp. 35–39, 2006.

[200] Wieczorek W., Unold O.: Use of a novel grammatical inference approach in classification of amyloidogenic hexapeptides, *Computational and Mathematical Methods in Medicine*, vol. 2016, 1782732, 2016. doi: 10.1155/2016/1782732.

[201] Wistrand M., Sonnhammer E.L.: Improving profile HMM discrimination by adapting transition probabilities, *Journal of Molecular Biology*, vol. 338, pp. 847–854, 2004. doi: 10.1016/j.jmb.2004.03.023.

[202] Won K.J., Hamelryck T., Prügel-Bennett A., Krogh A.: An evolutionary method for learning HMM structure: prediction of protein secondary structure, *BMC Bioinformatics*, vol. 8, 357, 2007. doi: 10.1186/1471-2105-8-357.

[203] Yandell M.D., Majoros W.H.: Genomics and natural language processing, *Nature Reviews Genetics*, vol. 3, pp. 601–610, 2002. doi: 10.1038/nrg861.

[204] Yokomori T., Kobayashi S.: Learning local languages and their application to DNA sequence analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1067–1079, 1998. doi: 10.1109/34.722617.

[205] Yoon B.J.: Hidden Markov models and their applications in biological sequence analysis, *Current Genomics*, vol. 10, pp. 402–415, 2009. doi: 10.2174/138920209789177575.

[206] Yoon B.J., Vaidyanathan P.P.: Structural alignment of RNAs using profile-csHMMs and its application to RNA homology search: Overview and new results, *IEEE Transactions on Automatic Control*, vol. 53, pp. 10–25, 2008. doi: 10.1109/TAC.2007.911322.

[207] Zadeh L.A.: Note on fuzzy languages, *Information Sciences*, vol. 1, pp. 421–434, 1969. doi: 10.1016/0020-0255(69)90025-5.

[208] Zehnder T., Benner P., Vingron M.: Predicting enhancers in mammalian genomes using supervised hidden Markov models, *BMC Bioinformatics*, vol. 20, 157, 2019. doi: 10.1186/s12859-019-2708-6.

[209] Zhang S., Borovok I., Aharonowitz Y., Sharan R., Bafna V.: A sequence-based filtering method for ncRNA identification and its application to searching for riboswitch elements, *Bioinformatics*, vol. 22, pp. e557–e565, 2006. doi: 10.1093/bioinformatics/btl232.

[210] Zhao Y., Hayashida M., Akutsu T.: Integer programming-based method for grammar-based tree compression and its application to pattern extraction of glycan tree structures, *BMC Bioinformatics*, vol. 11, S4, 2010. doi: 10.1186/1471-2105-11-s11-s4.

[211] Zhao Y., Hayashida M., Cao Y., Hwang J., Akutsu T.: Grammar-based compression approach to extraction of common rules among multiple trees of glycans and RNAs, *BMC Bioinformatics*, vol. 16, 128, 2015. doi: 10.1186/s12859-015-0558-4.

[212] Zou L., Wang Z., Wang Y., Hu F.: Combined prediction of transmembrane topology and signal peptide of β-barrel proteins: Using a hidden Markov model and genetic algorithms, *Computers in Biology and Medicine*, vol. 40, pp. 621–628, 2010. doi: 10.1016/j.compbiomed.2010.04.006.

Affiliations

Mariusz Flasiński
Jagiellonian University, Faculty of Management and Social Communication, Information Technology Systems Department, Cracow 30-348, Profesora Stanisława Łojasiewicza 4, Poland, mariusz.flasinski@uj.edu.pl

Received: 10.03.2024
Revised: 10.03.2024
Accepted: 10.03.2024

Computer Science • 25(1) 2024 https://doi.org/10.7494/csci.2024.25.1.5784

Mateusz Kocot
Krzysztof Misan
Valentina Avati
Edoardo Bossini
Leszek Grzanka
Nicola Minafra

USING DEEP NEURAL NETWORKS TO IMPROVE THE PRECISION OF FAST-SAMPLED PARTICLE TIMING DETECTORS

Abstract *Measurements from particle timing detectors are often affected by the time walk effect caused by statistical fluctuations in the charge deposited by passing particles. The constant fraction discriminator (CFD) algorithm is frequently used to mitigate this effect both in test setups and in running experiments, such as the CMS-PPS system at the CERN's LHC. The CFD is simple and effective but does not leverage all voltage samples in a time series. Its performance could be enhanced with deep neural networks, which are commonly used for time series analysis, including computing the particle arrival time. We evaluated various neural network architectures using data acquired at the test beam facility in the DESY-II synchrotron, where a precise MCP (MicroChannel Plate) detector was installed in addition to PPS diamond timing detectors. MCP measurements were used as a reference to train the networks and compare the results with the standard CFD method. Ultimately, we improved the timing precision by 8% to 23%, depending on the detector's readout channel. The best results were obtained using a UNet-based model, which outperformed classical convolutional networks and the multilayer perceptron.*

Keywords deep neural networks, timing detectors, diamond sensors, time series analysis, time walk correction, CERN, Precision Proton Spectrometer

Citation Computer Science 25(1) 2024: 43–61

1. Introduction

Precise time measurements of particles are crucial in many fields, including nuclear medicine and high-energy physics. Designing the most efficient method can be challenging, especially when the required precision is of the order of nanoseconds or picoseconds. Our research focuses on the detectors of the Precision Proton Spectrometer (PPS) [1], which is a subsystem of the Compact Muon Solenoid (CMS) [23] detector at CERN's Large Hadron Collider (LHC) [12]. At the LHC, protons and ions are accelerated to high energies and are then collided in dedicated interaction points. PPS detects and measures the kinematics of so-called forward protons, which are scattered to small angles after the interaction. Accurate calculation of the particle position and arrival time allows for the precise reconstruction of the particle trajectory and estimation of the interaction position which is crucial for the CMS-PPS subsystem.

The CMS-PPS system uses detectors installed on both sides of CMS, at a distance of approximately 220 m, to perform precise time measurements [7]. Each detector contains four detection planes with scCVD (single crystal Chemical Vapour Deposition) diamond sensors. When a charged particle passes through a sensor, it generates an electric analogue signal that is later amplified and digitised. In LHC Run 3[1], one of the digitisation techniques uses SAMPIC [11], a fast sampling ASIC (Application-Specific Integrated Circuit), on which we focus in our work. The chip samples and digitises the signal every 156.25 ps. Proper online or offline[2] analysis of these data including a multi-step preprocessing and filtering procedure can be used to compute the particle arrival time with high precision. The data flow during the analysis is depicted in Figure 1.

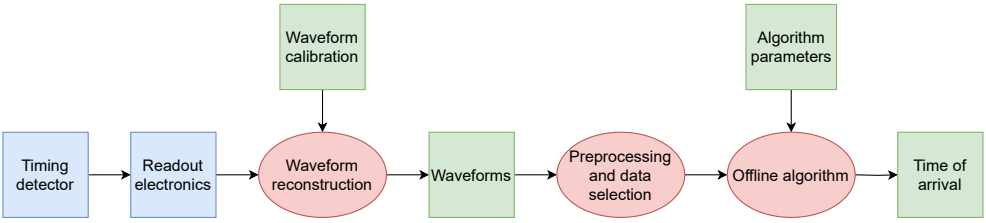


Figure 1. Data flow diagram for the analysis of the timing data (blue: electronics, green: data, red: digital algorithm). Multiple algorithms can be used to retrieve the time of arrival from the waveform data. In this research we focus on digital algorithms working in the offline mode.

The accuracy of the arrival time measurement is impacted by two main factors: jitter and the time walk effect. In our work, we focus on minimising the impact

[1]The operating period of the LHC, which started in 2022.

[2]The word 'offline' in this article is used to describe an algorithm working some time after data acquisition, contrary to online algorithms which work in the software or analogue pipeline straight after the data have been acquired.

of these components on the data from PPS sensors. Jitter is caused by the noise from the signal amplifier. Time walk is the dependence of the measured time on the signal amplitude. It is caused by statistical fluctuations of the charge released in a sensor by a passing particle. This leads to detecting signals with variable amplitudes. Signals with larger amplitudes cross a given threshold earlier than signals with smaller amplitudes. An example of measurements affected by the time walk effect is provided in Figure 2.

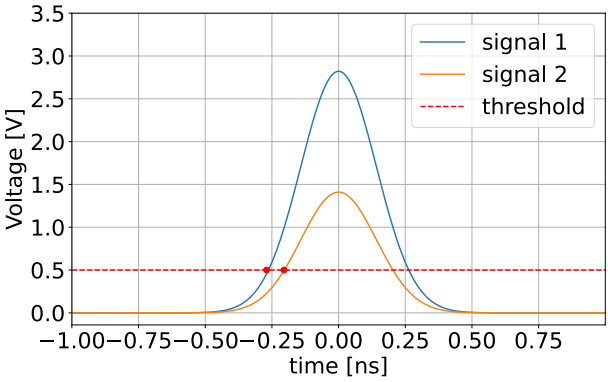


Figure 2. Time walk error illustrated as a difference in the threshold crossing times between two signals with the same shapes but different amplitudes

The constant fraction discriminator (CFD) is currently used at CMS-PPS to reduce the impact of the time walk effect and extract the time of arrival timestamps. The CFD is an analytical algorithm and does not use all available samples in a time series. Furthermore, the quality of its results is substantially reduced by the presence of noise and waveform irregularities. To address this issue, we propose a solution that utilises a deep neural network. This network can predict the arrival time of a particle from a sampled time series by using all available samples in a waveform.

2. State of the art

Various digital techniques are used to obtain the time of arrival. The classical approach is to use one of the multiple analytical methods. Following the recent trends, machine learning techniques are gaining popularity in this domain, too. This section outlines both of these strategies.

2.1. Analytical approaches

The simplest analytical method is the fixed threshold, which extracts the timestamp as the time of crossing a threshold fixed at a specific voltage. The main flaw of the fixed threshold is not taking the time walk effect into consideration at all.

The most common method used to mitigate the time walk effect is the normalised threshold algorithm, often referred to as the constant fraction discriminator (CFD). This algorithm normalises the waveform amplitude and then applies a fixed threshold. Other techniques include using the signal maximum as the timestamp or extracting only two timestamps and using the time over threshold (TOT) method [5, 9].

Due to its simplicity and relatively high performance, the constant fraction discriminator is considered the most reliable choice [6]. It is used in both test setups and running experiments, such as the CMS-PPS system at the LHC. Originally, the CFD was devised as an analogue device. However, we use it as an offline algorithmic solution to measure the arrival time of a particle given a very fast electrical pulse. By mitigating the error introduced by the time walk effect, the CFD allows for very accurate timing measurements. Given the excellent properties of the CFD and its common usage in the field, we selected this algorithm as the baseline for our numerical experiments.

2.2. Machine learning methods

Machine learning techniques are widely used in high energy physics. Common use cases include monitoring data quality by identifying outliers [2] and particle track reconstruction [19]. The short execution time of machine learning methods makes them useful for the high level trigger reconstruction task [18].

Although some supervised machine learning techniques, specifically deep neural networks, show promising results in time series analysis and timestamp prediction [21], they are seldom used to predict the time of arrival and have never been utilised for this purpose in the CMS-PPS subsystem.

The most extensive tests of neural networks in the domain of computing the time of arrival have been performed for MRPC (Multigap Resistive Plate Chamber) detectors. The research showed that multilayer perceptrons, LSTM (Long Short-Term Memory) recurrent neural networks or their combinations can be successfully used with the signals from a particle timing detector [26,27]. The interest is high in medical applications, too. Various convolutional architectures, mainly UNet-based, are used to calculate the time of flight in PET detectors [4] and to tag ECG diagrams [20, 28]. While these problems are different from the one discussed in this paper, they still require time series tagging, which is at the core of our problem.

3. Dataset

This section describes the data source and preprocessing steps required to construct the dataset used in this work. We also provide a detailed description of the version of the CFD algorithm, which is used during the data preprocessing procedure.

3.1. Data source

We constructed a dataset using the data acquired at the test beam facility in the DESY-II synchrotron in 2020 [8]. The facility hosted the PPS diamond timing detectors, as well as an MCP-PMT (Microchannel Plate Photomultiplier Tube) detector. The sensors were connected to the SAMPIC readout chip. The voltage time series sampled by SAMPIC had a fixed length of 64 samples within a 10 ns time window. Typically, the time window was long enough to capture the entire MCP signal. However, with diamond sensors, the signal is typically longer, with a wider trailing edge compared to the leading edge. As a result, in most cases, 10 ns was enough to fully capture only the leading edge of a signal.

The expected precision of the PPS diamond sensors was 50–100 ps, while the MCP timing precision was measured to be around 10 ps [8]. Considering its performance, MCP readouts were a perfect source of ground-truth information for our experiments. We present examples of waveforms acquired using the MCP and a diamond sensor in Figure 3.

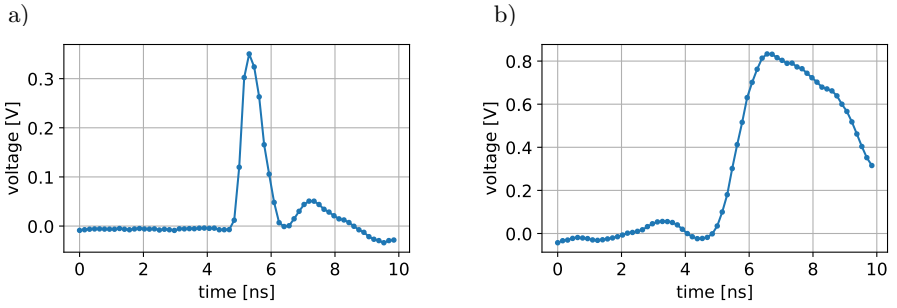


Figure 3. Example waveforms from the MCP (a) and a diamond detector (b)

3.2. Preprocessing

Multiple preprocessing steps were required for the data acquired from the DESY beam. Firstly, the samples were inverted to compensate for negative signals. This operation resulted in a rising edge in a signal indicating a particle.

The next preprocessing step was to eliminate noisy and so-called saturated events. Noisy signals are too weak to analyse, while saturated events occur when the voltage exceeds the amplifier's dynamic range, resulting in the signal being capped at a certain level. Examples of such events are shown in Figure 4. This means that the true amplitude cannot be easily retrieved from the signal, and the CFD cannot produce accurate reference values. We excluded noisy and saturated events from the analysis to focus solely on the comparison between deep learning models and the CFD.

The data filtering was done mainly using amplitude histograms. Amplitude (i.e. maximum voltage) was plotted on a histogram for each event. Typically, noisy events appear on the left side of the histogram, while saturated events appear on the right

side. Therefore, it is simple to filter out such events by selecting minimum and maximum amplitudes and discarding any events that fall outside this range. The minimum and maximum cuts were determined manually by analysing the histogram shapes. Figure 5 shows the maximum voltage histograms for the MCP and two selected diamond detectors. Due to its excellent waveform quality and low ratio of saturated events (visible as a small peak on the right side of its histogram), MCP did not require filtering.

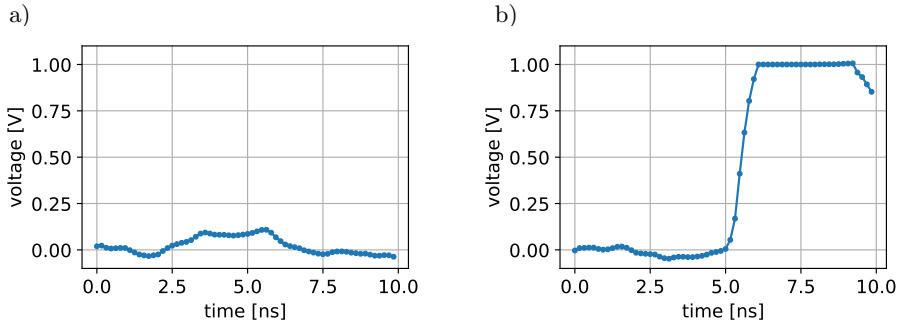


Figure 4. Examples of noisy (a) and saturated (b) events acquired using the diamond sensor

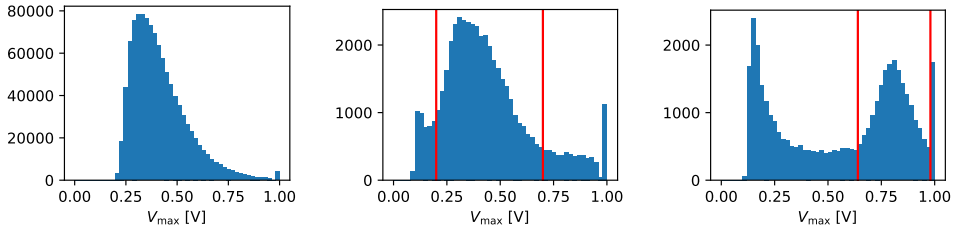


Figure 5. Maximum voltage histograms for MCP (left) and two selected diamond detectors (middle and right). The red lines on the histograms of the diamond detectors indicate the minimum and maximum amplitude cuts. It is important to note that the histograms look different for each detector, and therefore, it was necessary to find the amplitude cut values separately for each of them.

In the original dataset, the distribution of signal rising edge timestamps was centred around a single value in the middle of the time window. Using such a dataset would make it highly likely for the networks to overfit and collapse the predictions to the same timestamp for any input data. To avoid this, the signals were trimmed from 64 to 48 samples by removing 16 samples from the edges of the window. Specifically, up to 16 first samples were cut from each signal, and only the next 48 samples were kept. The number of first samples to drop was chosen randomly to smear the timestamp distribution.

Another issue was the fact that the waveforms had varying baselines and amplitudes. They had to be normalised in order to be properly processed by the neural networks. At first, the baseline was calculated as the mean value of the first 20 samples. Then, it was subtracted from the voltage values in the time series. Afterwards, the waveforms were divided by their maximums to normalise the amplitudes. The same steps are used in our version of the CFD algorithm and are visualised in Figure 7a.

3.3. Final dataset

In order to train the neural networks, we needed both time series and ground-truth (reference) time. Therefore, we constructed a dataset that only included events with corresponding readouts from both the MCP and a diamond sensor. The ground-truth timestamps were obtained from the MCP signals using the CFD method explained below. Due to the high quality of the MCP measurements, this approach was sufficient and provided the necessary timing precision for the training process. Figure 6 presents a normalised waveform from the final dataset, along with a corresponding MCP signal and the reference timestamp. The slight difference between the reference time shown on the MCP and diamond detector waveforms is due to the relative difference in the times of the first samples of both signals. This difference must be taken into account in the calculations to avoid bias.

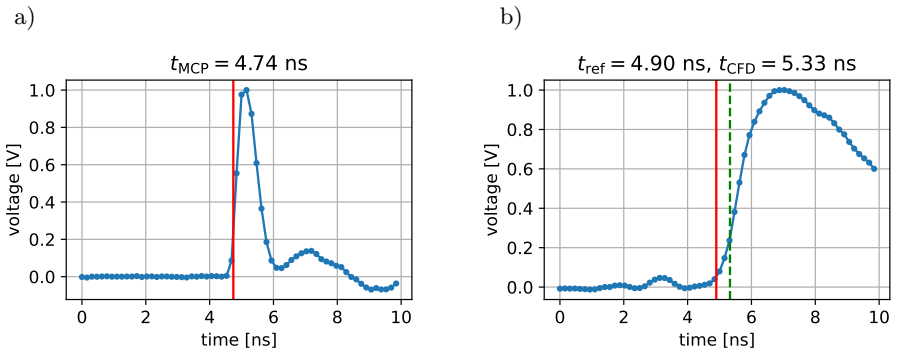


Figure 6. Example from the final dataset: a) waveform from MCP. t_{MCP} (red line) is computed using the CFD; b) waveform from a diamond detector. t_{ref} (red line) is the neural network's reference time computed using the CFD for the MCP waveform. The green dashed line represents the time computed using the CFD with the waveform from the diamond detector (t_{CFD}). It is not included in the dataset and is shown only for visualisation purposes

The final dataset consisted of approximately 500,000 waveform entries and their corresponding reference timestamps. After removing irrelevant information, performing data filtering and preprocessing, the size of the dataset was reduced from around 5.5 GB to 126 MB.

3.4. Constant fraction discriminator

The CFD algorithm used in this research is a modified version of the normalised threshold algorithm.

First, we normalise a time series using the same strategy as during data preprocessing, which involves baseline subtraction and division by the amplitude. Next, we calculate the time of arrival as the moment when the series crosses a chosen voltage threshold. To determine the exact timestamp, we apply a linear interpolation between the point before and the point after the threshold crossing. The chosen threshold is a fraction of the normalised amplitude, which ensures that the crossing point's dependence on the amplitude is removed. Figure 7 illustrates these steps.

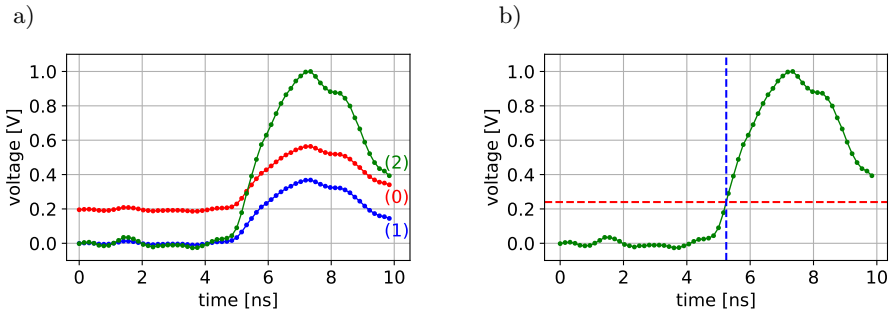


Figure 7. Depiction of the CFD algorithm: a) (0) before normalisation, (1) baseline subtraction, (2) division by maximum; b) when the normalisation is done, the timestamp can be found using the fixed threshold algorithm. The threshold chosen for this example is arbitrary

4. Network architectures

Our goal was to improve the timing precision using deep neural networks. We aimed to demonstrate that our methods achieve better results in terms of particle arrival time precision than the CFD. We started from a multilayer perceptron (MLP) and progressively increased the overall complexity of the network structure by using regular convolutional architectures and UNet-based [17] networks. We ran a hyperparameter tuning algorithm for each network type and selected the best candidates. Below, we briefly describe the training configuration, tuning method and hyperparameter options. We also depict the best-performing models.

4.1. Training configuration

We selected a single detector readout channel, i.e. a single diamond detector, for our primary tests. After preprocessing, we obtained 15,675 and 3,919 entries in the training and test sets, respectively. However, at this point, we left the test set for the final performance assessment. The models were trained using the Adam [15] optimiser with an adaptable learning rate which was reduced on learning curve plateaus.

As the output of the MCP and regular convolutional models was just a single number (the predicted timestamp), their metric was the squared error between the predicted and ground-truth values. In the case of the UNet-based model, it was trained to output a heatmap, so its error was calculated as the mean squared error between the predicted heatmap and the ground-truth one. The ground-truth heatmap was generated as a Gaussian with the mean at the true timestamp and a small standard deviation of one sampling step, i.e. 156.25 ps, following [25]. The final UNet timestamp could be retrieved as the mean of a Gaussian fitted to the output vector. The training process stopped when no improvements in the loss function were observed, following the early stopping method. This aimed to minimise the impact of overfitting [10].

4.2. Hyperparameter tuning procedure

Our hyperparameter tuning procedure consisted of two steps. First, we used a tuner algorithm to select the most promising models. Then, we performed cross-validation to determine the best one. The entire procedure utilised only the training set, with the same train-validation splits for every network type. We reserved the test set for final performance estimation.

For the first step, we chose one of the most common hyperparameter tuners, KerasTuner [16]. It is capable of selecting the top-N best models given an optimisation algorithm. We chose the Bayesian optimiser, which is an improved version of the grid search and random search algorithms. It estimates the loss function versus the hyperparameter values, and samples the hyperparameter sets according to that distribution. For each network type, we ran 40 iterations of KerasTuner, testing 40 different hyperparameter sets. Each set was trained using 80% of the original training set, while the remaining 20% was used for validation. To improve the quality of the results and filter out unstable models, we used two executions per trial, meaning that the result of a model was calculated as an average of the loss values from these two, separate executions of training and validation.

In the second phase, we used the top 5 models outputted by the tuner and performed 5-fold cross-validation using only the training set. The folds were consistent across all network types, and each fold value was an average of three trials. The final model for a given network type was chosen based on the mean and standard deviation of the cross-validation results.

The hyperparameter tuning procedure was run on the High-Performance Computing GPU cluster. The computations took from one to four hours, depending on the network architecture.

4.3. Multilayer perceptron

The main goal of tuning the multilayer perceptron (MLP) architecture was to find the optimal number of hidden layers and neurons. To avoid using a separate hyperparameter for the number of neurons in each layer, we assumed that either the same

number was used or that the number was divided by 1.5 or 2 for every consecutive dense layer. The final layer always had only one neuron, since it was the output of the model and represented the predicted timestamp.

In addition to these parameters, we also tested the effects of adding batch normalisation [14] and dropout [22]. Batch normalisation could be added either after every dense layer or not at all, and could also be added independently after the input layer. Finally, dropout was set to either 0, 0.2, or 0.5. An activation function was applied after every dense layer, except for the last one, and was fixed to ReLU [13].

The best models returned by the tuning procedure almost always included batch normalisation after every dense layer and the input but did not use dropout. There were no clear patterns for the rest of the hyperparameters. The final MLP architecture is shown in Figure 8.

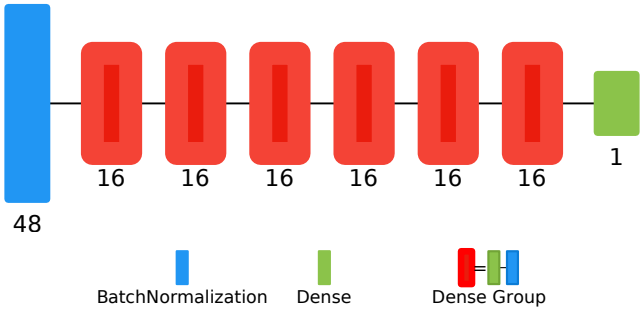


Figure 8. Optimal MLP model [3]

4.4. Convolutional network

The convolutional neural network (CNN) consisted mainly of convolutional layers, which are commonly used in image and time series processing tasks. Unlike dense layers, convolutional layers can learn the relations between neighbours, such as time series samples located next to each other.

Typically, the number of filters increases with each consecutive convolutional layer. In our case, it was multiplied by 2. However, up to three convolutional layers could be used in sequence before increasing the filter count. We refer to this group as a convolutional block. The number of blocks was another hyperparameter, ranging from 1 to 4. Only the number of filters in the first block was optimised, as the numbers in the following blocks could be inferred from the number in the first block. All convolutional layers had kernels of small size: 3. A single dense layer was placed at the end of the network so that the network could output a single number. A small MLP could be inserted between the convolutional part of the network and the final dense layer, parametrised similarly to the full MLP architecture. However, its depth was limited to 3.

In addition to the core layers, batch normalisation was parametrised similarly to the MLP. Dropout could be applied after each dense layer in the MLP part. For convolutional layers, we used spatial dropout [24] instead of regular dropout. Its rate could be set to 0.0, 0.1, or 0.2.

Batch normalisation was used in all of the models returned by the tuner, while the MLP dropout was always set to 0.0. No visible patterns were observed for other hyperparameters. The final convolutional architecture is shown in Figure 9.

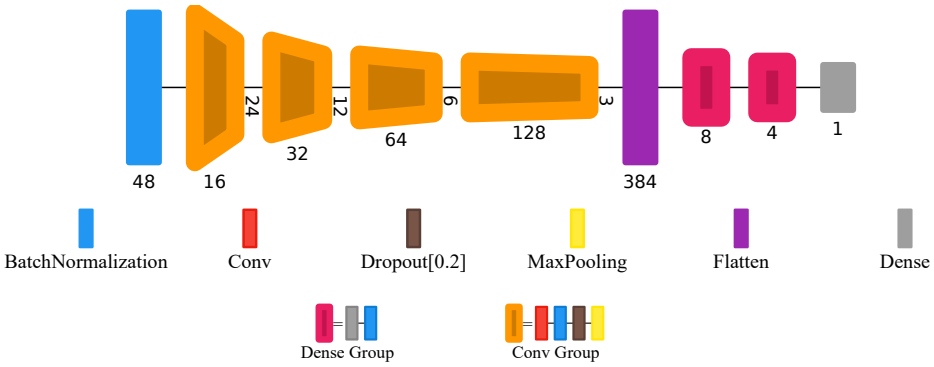


Figure 9. Optimal CNN model [3]

4.5. UNet

The last architecture we used, UNet, is characterised by the U shape of its architecture. It is composed of an encoder and a decoder. The encoder extracts relevant features from the input, while the decoder uses those features to build a vector of the input shape and highlight relevant spots, such as predicted timestamps in time series processing. The UNet architecture takes advantage of skip connections to amplify the importance of initial features in the decoder. Thanks to the segmentation and noise reduction capabilities of UNet, we expected it to be a good candidate for our task.

The main hyperparameter for our model of the network depth, measured in UNet blocks. The encoder and decoder were symmetrical and contained the same number of blocks. An encoder block consisted of one to three convolutional layers followed by a max pooling layer. A decoder block started with deconvolution, which we implemented through upsampling and a convolutional layer with a kernel size of one (all other convolutional layers had a kernel size of 3). The output of a decoder block was concatenated with the output from the corresponding block in the encoder through a skip connection. Finally, one to three convolutional layers were used. As before, batch normalisation and spatial dropout could be added after every convolutional layer with a kernel size of 3. Batch normalisation could also be used after input. The final UNet architecture is depicted in Figure 10.

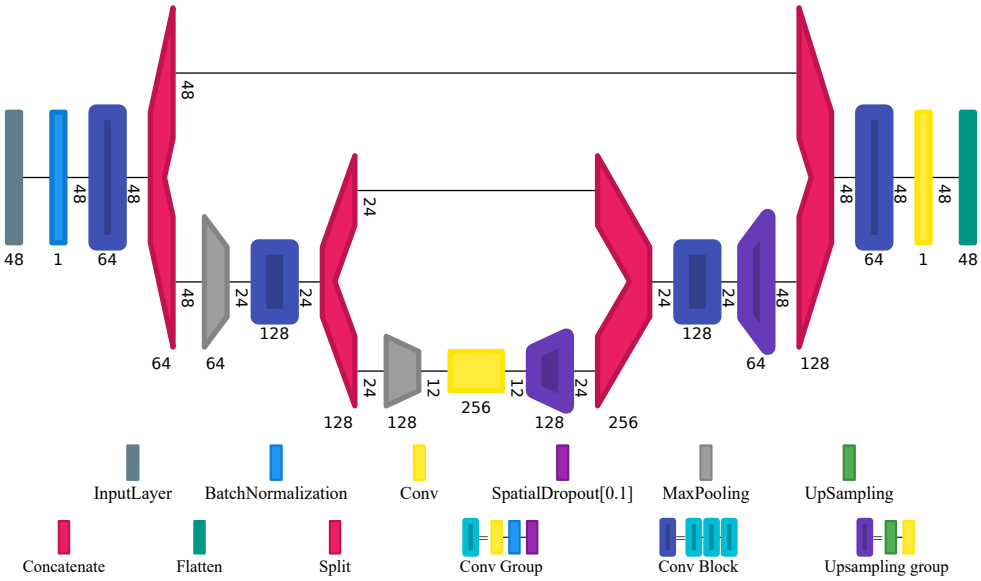


Figure 10. Optimal UNet model [3]

5. Results

This section begins with a description of the method used to assess the timing precision of either the CFD or a deep neural network. Then, we compare the results obtained with the CFD to those obtained with the best neural network using data from a single readout channel of the detector. We also provide a description of other performance tests we performed, including the tests on other readout channels.

5.1. Precision assessment method

To measure the time precision of a detector, a typical method is to compare its measurements with those of a 'reference' detector that has much better time precision. The reference detector is placed on the same beam line to detect the same particles. The mean of the differences between the tested detector and the reference detector represents a constant offset. Although neural networks can learn to have a mean close to zero, the CFD mean is expected to be shifted due to the method's inability to adjust to inconsistent signal characteristics. The precision of the time measurement is represented by the standard deviation of the differences.

In our case, MCP served the role of the reference detector. What is more, to reduce tail effects, we fitted a Gaussian curve to the histogram of the time differences and used the standard deviation of the Gaussian as the precision measure (as shown in Figure 11).

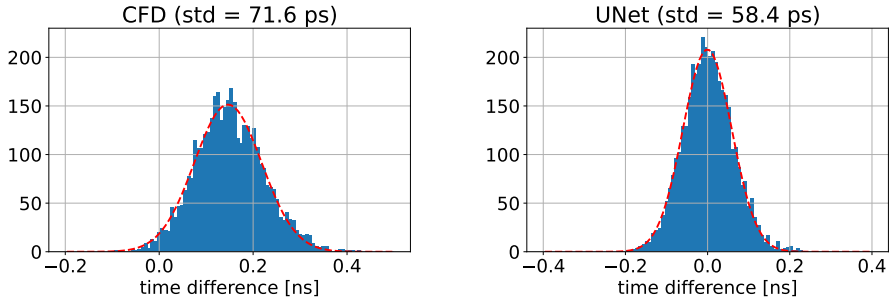


Figure 11. Difference histograms for the CFD and our best-performing, UNet-based model

5.2. The optimal architecture

We performed cross-validation on the best models, one from each network type. Instead of computing loss values, we used the evaluation method described above to compute the results. The computations were performed only on the training portion of the dataset. The results are shown in Table 1. As expected, UNet had the smallest (the best) average precision value. It was also the most complex model with the biggest number of parameters. Interestingly, the model had more parameters than the number of training samples. This is typical for neural networks as they often use more parameters than the minimum required number. While this can make the network susceptible to overfitting, with proper training, overfitting can be avoided. The early stopping method we employed is the best approach to address this issue. Additionally, spatial dropout was applied to improve performance at the expense of further increasing the number of parameters. Surprisingly, the MLP model was the most stable, with the smallest standard deviation of results for each fold.

Table 1

Comparison of the precisions achieved by the optimal models in the cross-validation procedure. In addition to the cross-validation scores, the number of parameters used by each network is reported, too

Architecture	Mean [ps]	Std [ps]	Parameter count
MLP	63.90	0.85	2737
CNN	62.83	1.34	36,865
UNet	60.71	1.19	456,965

To evaluate the final performance of our solution, we used the test set which was composed of data from the same readout channel as the training set. Figure 11 shows the difference histograms for the CFD and our best-performing neural network. The network's histogram is visibly narrower, indicating better precision.

5.3. Adjusting the data to the LHC conditions

Due to limited available bandwidth in the PPS setup at the LHC, the SAMPIC time series consist of only 24 samples. To validate the networks under these conditions, we trimmed the original time series from 64 to 24 samples. First, we smeared the timestamp distribution by randomly removing up to 10 samples from the beginning of the series, retaining only the next 56 samples. We then selected the 24 samples from the middle to ensure that the most important part of the signal was preserved. We made slight adjustments to the network architectures to accommodate the smaller input size (24 instead of 48). We performed the same tests as before and obtained similar results. We were able to improve the precision from 73.3 ps to 62.1 ps standing for 15%, which is promising for using deep learning with LHC data.

5.4. Tests on many channels

In the previous sections, we only used data from a single detector channel. However, we also tested data from other channels. We first trained the optimal network on each channel separately and then tested it on the same channel which was used for training. We achieved precision improvements ranging from 8% to 23% compared to the CFD. We also investigated if the network could be trained on one channel and tested on another, or even trained on all available data while maintaining the train-test split. We present the results for selected, representative channels in Table 2.

Table 2

Precision improvements with respect to the CFD obtained with many detector channels involved [%]. We selected seven, representative channels and highlighted the best precision for each one

Training channel	Test channel						
	10 [%]	16 [%]	17 [%]	22 [%]	25 [%]	27 [%]	31 [%]
10	**13**	10	13	7	–1	–23	–7
16	6	**23**	16	9	–22	–9	3
17	7	17	**19**	9	–3	8	–6
22	4	14	–4	11	–84	–51	7
25	4	4	7	4	**12**	8	–4
27	–13	–10	4	–16	4	**19**	–17
31	2	13	10	7	–7	–7	**16**
all	8	22	14	**12**	9	17	14

These results show that, typically, in order to achieve the highest precision for a given channel, the network needs to be trained using data from that channel specifically. If trained on one channel and tested on another, the network might perform worse than the CFD, resulting in negative entries in Table 2. This shows that even though the same sensor is used, the collected data differ significantly between channels. Channel 22 is the only channel for which the network trained on all channels

was able to achieve slightly better precision than the network trained on that specific channel. This may be due to an unfavourable train-test split for the channel 22 data. We present the improvements for all the channels we explored using the networks trained on the particular channels in Figure 12. The data were available for channels from 8 to 31. We do not report the improvements for channels 12 and 15, as the waveforms were too noisy.

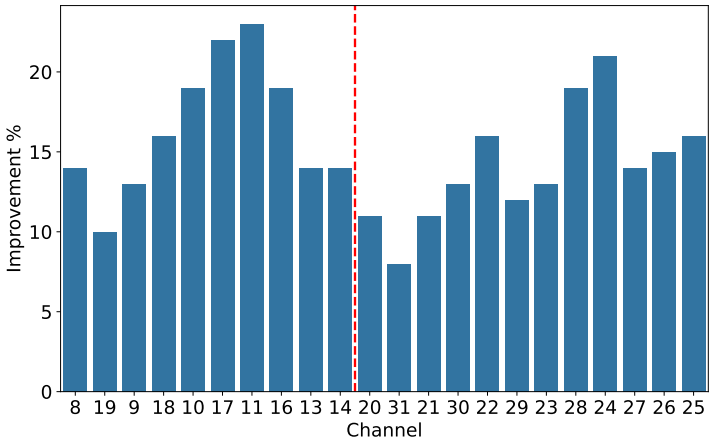


Figure 12. Improvements with respect to the CFD for the channels we explored. The channel order follows the physical set-up of the test beam experiment. The red dashed line divides the channels from two separate planes of diamond sensors used in the experiment

It is worth noting that we did not test the network on data from all available channels at once. This would be difficult due to the different mean values of the difference histograms in various channels. Instead of a single Gaussian, we would have a group of smeared Gaussians, which would make it impossible to retrieve the true detector precision. Therefore, even when the network was trained using data from all channels, we tested it separately for each channel.

6. Conclusion

We demonstrated that deep neural networks can be used to compute the time of arrival of particles taking samples voltage signals at input. In fact, these networks can improve timing precision compared to the most commonly used algorithm, the CFD. In the base numerical experiment, we were able to improve precision by 17%. It is a significant value considering that we did not make any modifications the detector setup, but just used a different algorithm for computing the time of arrival. Other readout channels also showed improvements ranging from 8% to 23%. We found that networks based on the UNet architecture yield the best results among the models we investigated. However, we did not test recurrent networks, which are also expected to

perform well in this kind of problem. We leave that for further research. Nonetheless, even the simplest network architecture we tested – MLP – enabled us to calculate the arrival time with noticeably better precision than the CFD.

Neural networks have a wide range of applications in high energy physics. For example, some types of neural networks are used to evaluate the quality of waveform-like experimental data, thereby improving the detection of outliers and bad data [2]. Additionally, the resolution of pattern recognition algorithms is improved for detectors with complex geometries [19]. This work contributes to the evaluation of neural networks in high-energy physics. The method proposed in this manuscript broadens their applicability and increases the precision of timing detectors. Deep neural networks have been tested in similar applications. It has been shown that they can be used in the prediction of arrival time [26, 27], or more generally in the annotation of time series [20]. This proves that our findings are not just a coincidence. However, it is worth noting that neural networks can achieve high accuracy only on data similar to the training dataset.

The method described in [26] differs from our approach in terms of the detector architecture and the usage of simulated data in network training. In contrast, our work relies entirely on experimental data and a more precise source of reference data in the form of an MCP-PMT detector.

It is important to note that our reference data were not flawless. The MCP precision was assessed to be 10 ps, which is much better than that of diamond sensors (about 50–100 ps). Nevertheless, MCP signals are not perfect and introduce a small degree of uncertainty. Using the CFD further worsens the reference precision. The resulting error is random and can cause some events to contradict each other during neural network training. For instance, reference timestamps may vary for waveforms that look identical. As a result, the final precision of the neural network was negatively impacted.

In addition to the time of arrival study, the procedure developed in this research using KerasTuner to tune hyperparameters and find the optimal network architecture has been successfully applied in ongoing studies to improve the precision of timing computations in the CMS-PPS subsystem at the LHC.

The Large Hadron Collider (LHC) restarted in 2022 and is producing data in a format similar to that discussed in this article. The SAMPIC readout board saves the full waveform in the raw data stream, which can be subject to further analysis using the method presented in this manuscript. Although the CFD is still commonly used for timing measurements, it may be replaced by neural networks. Lack of the MCP in the LHC setup poses a problem in terms of the reference data acquisition, but the work on finding another way is ongoing.

The inference from the neural network has a very low demand for the CPU time (order of 10 milliseconds), making it well-suited for online processing, such as in the high-level trigger reconstruction chain. The data can be processed in batches, enabling efficient parallel processing of a large number of events.

Acknowledgements

The project was partially funded by the Polish Ministry of Education and Science, project 2022/WK/14. This research was supported in part by PL-Grid Infrastructure. The numerical experiment was possible through computing allocation on the Ares system at ACC Cyfronet AGH under the grant plgccbmc11.

References

[1] Albrow M., Arneodo M., Avati V., Baechler J., Cartiglia N., Deile M., Gallinaro M., *et al.*: CMS-TOTEM Precision Proton Spectrometer. Technical design report. CMS 13, Technical design report. TOTEM 3, 2014. https://cds.cern.ch/record/1753795/.

[2] Azzolin V., Andrews M., Cerminara G., Dev N., Jessop C., Marinelli N., Mudholkar T., *et al.*: Improving data quality monitoring via a partnership of technologies and resources between the CMS experiment at CERN and industry, *EPJ Web of Conferences*, vol. 214, 01007, 2019. doi: 10.1051/epjconf/201921401007.

[3] Bäuerle A., Onzenoodt van C., Ropinski T.: Net2Vis – A Visual Grammar for Automatically Generating Publication-Tailored CNN Architecture Visualizations, *IEEE Transactions on Visualization and Computer Graphics*, vol. 27(6), pp. 2980–2991, 2021. doi: 10.1109/TVCG.2021.3057483.

[4] Berg E., Cherry S.R.: Using convolutional neural networks to estimate time-of-flight from PET detector waveforms, *Physics in Medicine & Biology*, vol. 63(2), 02LT01, 2018.

[5] Berretti M., Bossini E., Minafra N.: Timing performances of diamond detectors with Charge Sensitive Amplifier readout, Technical report, CERN-TOTEM-NOTE-2015-003, 2015. https://cds.cern.ch/record/2055747.

[6] Bossini E.: *Development of a Time Of Flight diamond detector and readout system for the TOTEM experiment at CERN*, Ph.D. thesis, INFN, Siena, 2016. https://cds.cern.ch/record/2227688. CERN-THESIS-2016-137.

[7] Bossini E.: The CMS Precision Proton Spectrometer timing system: performance in Run 2, future upgrades and sensor radiation hardness studies, *Journal of Instrumentation*, vol. 15(05), C05054, 2020. doi: 10.1088/1748-0221/15/05/C05054.

[8] Bossini E., Figueiredo D.M., Forthomme L.M., Garcia Fuentes F.I.: Test beam results of irradiated single-crystal CVD diamond detectors at DESY-II, Technical reports, CMS-NOTE-2020-007, CERN-CMS-NOTE-2020-007, 2020.

[9] Breton D., De Cacqueray V., Delagnes E., Grabas H., Maalmi J., Minafra N., Royon C., Saimpert M.: Measurements of timing resolution of ultra-fast silicon detectors with the SAMPIC waveform digitizer, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 835, pp. 51–60, 2016. doi: 10.1016/j.nima.2016.08.019.

[10] Caruana R., Lawrence S., Giles C.: Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In: T. Leen, T. Dietterich, V. Tresp (eds.), *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, pp. 402–408, MIT Press, 2001.

[11] Delagnes E., Breton D., Grabas H., Maalmi J., Rusquart P., Saimpert M.: The SAMPIC waveform and time to digital converter. In: *2014 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pp. 1–9, IEEE, 2014. doi: 10.1109/nssmic.2014.7431231.

[12] Evans L., Bryant P.: LHC machine, *Journal of Instrumentation*, vol. 3(08), S08001, 2008. doi: 10.1088/1748-0221/3/08/S08001.

[13] Glorot X., Bordes A., Bengio Y.: Deep sparse rectifier neural networks. In: *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTAT)*, pp. 315–323, JMLR Workshop and Conference Proceedings, 2011.

[14] Ioffe S., Szegedy C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pp. 448–456, JMLR.org, 2015.

[15] Kingma D.P., Ba J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:14126980*, 2014. doi: 10.48550/arXiv.1412.6980.

[16] O'Malley T., Bursztein E., Long J., Chollet F., Jin H., Invernizzi L., *et al.*: KerasTuner, https://github.com/keras-team/keras-tuner, 2019.

[17] Ronneberger O., Fischer P., Brox T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: N. Navab, J. Hornegger, W. Wells, A. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, vol. 9351, pp. 234–241, Springer, Cham, 2015. doi: 10.1007/978-3-319-24574-4_28.

[18] Schefer M.M.: *Machine Learning Techniques for selecting Forward Electrons* ($2.5 < |\eta| < 3.2$) *with the ATLAS High Level Trigger*, Technical report, ATL-DAQ-PROC-2023-001, 2023. https://cds.cern.ch/record/2851302.

[19] Shlomi J., Battaglia P., Vlimant J.R.: Graph neural networks in particle physics, *Machine Learning: Science and Technology*, vol. 2(2), 021001, 2020. doi: 10.1088/2632-2153/abbf9a.

[20] Sodmann P., Vollmer M., Nath N., Kaderali L.: A convolutional neural network for ECG annotation as the basis for classification of cardiac rhythms, *Physiological Measurement*, vol. 39(10), 104005, 2018.

[21] Song X., Liu Y., Xue L., Wang J., Zhang J., Wang J., Jiang L., Cheng Z.: Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model, *Journal of Petroleum Science and Engineering*, vol. 186, 106682, 2020.

[22] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R.: Dropout: a simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research*, vol. 15(1), pp. 1929–1958, 2014.

[23] The CMS Collaboration: The CMS experiment at the CERN LHC, *Journal of Instrumentation*, vol. 3, S08004, 2008. doi: 10.1088/1748-0221/3/08/S08004.

[24] Tompson J., Goroshin R., Jain A., LeCun Y., Bregler C.: Efficient object localization using convolutional networks. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 648–656, 2015. doi: 10.1109/cvpr.2015.7298664.

[25] Tompson J.J., Jain A., LeCun Y., Bregler C.: Joint training of a convolutional network and a graphical model for human pose estimation, *Advances in Neural Information Processing Systems*, vol. 27, 2014. doi: 10.48550/arXiv.1406.2984.

[26] Wang F., Han D., Wang Y.: Improving the time resolution of the MRPC detector using deep-learning algorithms, *Journal of Instrumentation*, vol. 15(09), C09033, 2020. doi: 10.1088/1748-0221/15/09/C09033.

[27] Wang F., Han D., Wang Y., Yu Y., Lyu P., Guo B.: The study of a new time reconstruction method for MRPC read out by waveform digitizer, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 954, 161224, 2020. doi: 10.1016/j.nima.2018.09.059.

[28] Zahid M.U., Kiranyaz S., Ince T., Devecioglu O.C., Chowdhury M.E., Khandakar A., Tahir A., Gabbouj M.: Robust R-Peak Detection in Low-Quality Holter ECGs Using 1D Convolutional Neural Network, *IEEE Transactions on Biomedical Engineering*, vol. 69(1), pp. 119–128, 2022. doi: 10.1109/tbme.2021.3088218.

Affiliations

Mateusz Kocot
AGH University of Krakow, Krakow, Poland, mateusz.kocot@cern.ch

Krzysztof Misan
AGH University of Krakow, Krakow, Poland, krzysztof.misan@cern.ch

Valentina Avati
AGH University of Krakow, Krakow, Poland, valentina.avati@cern.ch

Edoardo Bossini
INFN Sezione di Pisa, Pisa, Italy, edoardo.bossini@pi.infn.it

Leszek Grzanka
AGH University of Krakow, Krakow, Poland, grzanka@agh.edu.pl

Nicola Minafra
University of Kansas, Department of Physics and Astronomy, Lawrence, KS, United States, nicola.minafra@cern.ch

Received: 30.09.2023
Revised: 06.12.2023
Accepted: 07.12.2023

Computer Science • 25(1) 2024 https://doi.org/10.7494/csci.2024.25.1.5685

Bhupendera Kumar
Rajeev Kumar

GENERALIZING CLUSTERING INFERENCES WITH ML AUGMENTATION OF ORDINAL SURVEY DATA

Abstract *In this paper, we attempt to generalize the ability to achieve quality inferences of survey data for a larger population through data augmentation and unification. Data augmentation techniques have proven effective in enhancing models' performance by expanding the dataset's size. We employ ML data augmentation, unification, and clustering techniques. First, we augment the limited survey data size using data augmentation technique(s). Second, we carry out data unification, followed by clustering for inferencing. We took two benchmark survey datasets to demonstrate the effectiveness of augmentation and unification. The first dataset contains information on aspiring student entrepreneurs' characteristics, while the second dataset comprises survey data related to breast cancer. We compare the inferences drawn from the original survey data with those derived from the transformed data using the proposed scheme. The results of this study indicate that the machine learning approach, data augmentation with the unification of data followed by clustering, can be beneficial for generalizing the inferences drawn from the survey data.*

Keywords survey research, ordinal data, data augmentation, clustering, unification, generalization

Citation Computer Science 25(1) 2024: 63–93

1. Introduction

Surveys are the most popular form of data collection in organizational and behavioral research [5]. The areas of policy-making, higher education, health care, psychology, and market research are some of the ones that commonly use surveys [15]. Correctly processing survey data has become a major problem due to the vast range of applications. Minor survey data analysis may occasionally produce bizarre results. Therefore, the right analytical tools are necessary to derive relevant insights from survey data. However, the nature of the data and the application's goal significantly impact how reliable analysis tools are [38]. It attempts to comprehend a phenomenon by compiling feedback from a sizable population [5].

The standard methods for analysis in survey research are statistical modeling tools for finding survey error(s); this necessitates prior knowledge of the association between the outcomes and covariates [46]. Unfortunately, in complex real-world circumstances where these interactions may not be accessible, it is not always possible to satisfy the condition of understanding the relationship mapping between the outcomes and variables. More adaptable modeling strategies are necessary for these situations that do not call for relational mappings to be predefined. Complex circumstances can be better understood by building relational mappings based on the inherent properties of the data [24]. For example, grouping data points according to their natural proximity can help us better comprehend a phenomenon, like the behavior of a sampled population. Flexible modeling techniques must be used, and numerous data-related issues must be resolved to extract relevant and trustworthy insights from survey data. Unique qualities of survey data include variability, hierarchical linkages, and the importance of category names [43]. Depending on the degree of heterogeneity, the data may contain a variety of metrics, including binary, continuous, categorical, or their mixtures.

Most survey techniques involve using a single mode of data collection. In today's complex world, single-mode survey techniques may not be sufficient. For example, universities survey students to learn about their perspectives, interests, and behavior to better understand the factors that contribute most to their entrepreneurial aptitude; the survey data could be multi-modal.

To address this, researchers employ multiple surveys allowing diverse inferencing and catering to complex themes. These surveys offer a range of methods, including mathematical analysis and qualitative inference, to gather comprehensive data and insights that align with the research objectives and complexities of the survey topic [6]. Unification is a process of combining various data elements to create an arrangement that is logical and consistent enough to allow for the drawing of reliable inferences. Since it makes it feasible to combine and bring diverse pieces of knowledge into one coherent whole, unification is crucial for effective inferencing. It must include patterns or connections into a single structure. Furthermore, unification calls for considering relevant factors affecting the discovered patterns or correlations [42]. The success of unification is crucial for accurate inference.

In addition, sampling is always limited by size, yet it is expected that such limited sampling should lead to the views of the whole population [33]. Participants might differ in their perspectives. Ensuring the sample size is enough to include all relevant viewpoints while performing qualitative research. A limited sample size can have a better chance of finding a wide range of impressions and increasing the credibility of their inferences; there could be fewer conflicts. However, analyzing such a range of data presents formidable difficulties [34].

Additionally, survey data generated by web-based survey software frequently contains small ordinal measurements. According to the research, treating values on small ordinal scales as value-based is improper [47]. On the other hand, methodologies that make use of vectors with ordinal values typically outperform pattern-based analysis techniques [40, 41]. Such techniques, widely used by most survey inferencing tools and techniques, lead to arbitrary inferencing. Apart from the facts above about the reliable analysis of survey data, the flexible modeling techniques, and the use of vector techniques, the survey faces the challenge of gathering information from the intended audience. It is one of the main problems with online surveys. Online surveys frequently require greater response rates, which could result in sufficient sample size and skewed findings [2].

In this work, we use machine learning (ML) techniques, namely, data augmentation, to augment the survey size. ML is mostly data-driven [30]. To put it another way, it offers adaptable modeling methods that exclusively rely on the intrinsic properties of the data to make the connections between the data and the results. ML usage may open survey research to generalized predictive modeling, limited to determining population features from a sample of data [8, 9]. Data augmentation is a generalization technique that enriches and enhances the population size for proper inference. We expect that the inferencing of the limited survey should be that of the population [22].

In this work, we focus only on ordinal data. But, like in many surveys, there is also associated numerical type data. So, as a result, we apply the unification process, which appropriately converts numerical values to ordinal values. After this, we put the dataset into a machine-learning model, especially for clustering. By doing this, we got better clusters for finding the effecting features from each cluster. It means the formed clusters are effective. In the result section, we show the efficiency measures of the clustering and can find suitable and effective features.

Therefore, we address the following research questions (RQs) in this work:

RQ1: Does the limited survey sampling be extended through ML augmentation reflecting a more significant population's general opinion?

RQ2: Does the augmented data with unification and clustering yield proper inferences?

Regarding the RQs mentioned above, the research in this work examines the proper inferencing obtained through augmentation and unification. For this purpose, we took two case studies, one for finding the competency factors in university

entrepreneurs and the second for breast cancer prediction features. The research contributions of this work are summarized as:

- We employ data augmentation techniques on *limited* survey to yield the inferencing of larger population.
- We identify the generalized driving features to determine the significant factors for prediction.
- The proposed ML methods yield significant inferences for ordinal-type survey data for proper decision-making.

The paper is organized as follows: Section 2 describes the motivation behind developing ML-driven methods. Section 3 briefly reviews the literature highlighting the data augmentation techniques and unification's role in clustering survey data. In Section 4, we describe, in detail, the concept of our proposed methodology. The experimental setup with dataset description of survey data and the detailed corresponding results are presented in Section 5.1. Finally, we conclude the paper in Section 6. Table 1 lists the key abbreviations that comprise this paper.

Table 1
Abbreviations

Abbreviation	Description
ADASYN	ADAptive SYNthetic
CNFL	Categorical to Numerical Feature Learning
DAUG	Data AUGmentation
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNA	Deoxyribo Nucleic Acid
EM	Expectation Maximization
GA	Genetic Algorithm
GAN	Generative Adversarial Network
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristics
SMOTE	Synthetic Minority Over Sampling
SOM	Self Organising Map
UFDM	Unification For Data Modelling

2. Motivation

The survey of a limited population should reflect the opinion of a large population. The survey aims to gather perceptions and viewpoints that may be applied to a more significant population. A properly chosen sample population that reflects the large population in terms of the pertinent features must be used to do this. The techniques may be used to draw valid conclusions about the attitudes and actions of a larger population. We are using ML techniques. It should be generalized. Therefore, we use the augmentation technique, which is a generalization technique. Even then, if we get

fewer responses, it may give loosely prominent results. So we have to expand our number of responses through augmentation. We depict an example of this process. Here, we assume only three features (A1, A2, and A3) and responses from two categories (A and B). We suppose that a survey is conducted on a questionnaire, and based on this, there are three features to observe (A1, A2, and A3). The features will be selected from three for decision-making on the two categories of responses (A and B).

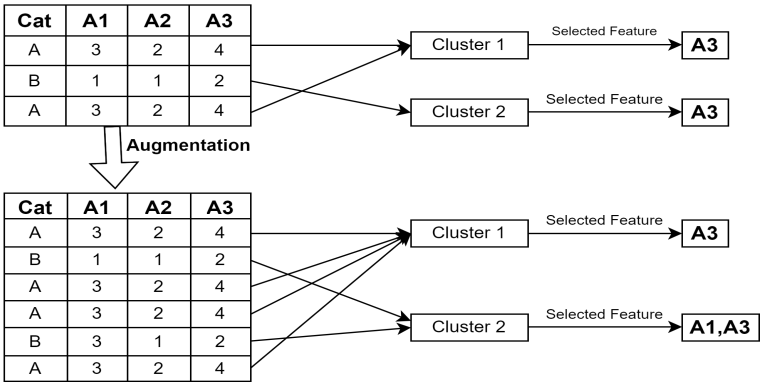


Figure 1. A Sample Example of Selected Features from a Survey Dataset

Figure 1 illustrates that in a small population size, before augmentation, the most governing feature for inferences through clustering is only A3. Considering the responses belong to two categories, A and B, they group into two clusters. A3 is most likely selected for inferencing from each cluster with maximum grading. On the other hand, after augmentation, the responses increased in number and were also grouped into two clusters. But this time, we got another feature A1 from cluster 2. There may be a possibility of getting all three features; this will be discussed in detail in the results section for limiting the selected features. Thus, we can get more generalized and precise inferences. Therefore, the inferences of ML techniques will be better suitable for this work.

3. Related work

3.1. Data augmentation in survey data

The fundamental objective of data augmentation is to create a productive and repeatable sampling method by adding concealed or unseen factors to the model. This technique gained prominence primarily in deterministic algorithms that aim to maximize likelihood functions or posterior densities with the expectation-maximization (EM) algorithm [16]. Constructing a data augmentation algorithm is somewhat of an art because data augmentation algorithms must be carefully developed for each model type [17]. Schliep and Hoeting introduced parameter-expanded data augmentation techniques to model ordinal data with the *probit* model. Specifically, the study

focused on implementing these algorithms for the probit linear mixed model in the context of spatially correlated ordinal response data. The researchers then demonstrated the applicability of the model by utilizing it to assess the biotic integrity of wetlands in Colorado [42].

Machine learning algorithms are typically evaluated based on their predictive accuracy. However, this approach may be unsuitable for imbalanced datasets where classes are not evenly represented or when the cost of different errors varies significantly. For instance, fraud detection often involves a class imbalance of 100 to 1, while other applications may have an imbalance of up to 1,00,000 to 1. Over-sampling techniques have been proposed to address this issue to balance the data. One approach involves creating synthetic examples of the minority class rather than simply over-sampling with replacement. This technique has been successful in handwritten character recognition, where operations like rotation and skew were used to perturb the training data and create additional examples. By generating synthetic examples, we can improve the training of machine learning algorithms on imbalanced data and ensure that the minority class is not overlooked. This approach can be precious in applications like fraud detection, where correctly identifying the minority class is critical. SMOTE (Synthetic Minority Over-sampling Technique) [12] demonstrates that a more effective classifier performance (in ROC space) can be achieved through a combination of our over-sampling method for the minority (abnormal) class and under-sampling for the majority (normal) class, compared to solely under-sampling the majority class.

The Adaptive Synthetic (ADASYN) [23] sampling approach has been developed to address these issues. The main idea behind ADASYN is to use a weighted distribution for different minority class examples based on their level of difficulty in learning. This means that more synthetic data is generated for minority class examples that are harder to learn than those that are easier to learn. As a result, ADASYN improves learning by reducing the bias introduced by class imbalance and adaptively shifting the classification decision boundary towards the difficult examples. Simulation analyses on several machine learning data sets have demonstrated the effectiveness of this approach across five evaluation metrics. The ADASYN sampling approach has emerged as a promising solution to this challenge. By generating synthetic data for minority class examples based on their level of difficulty in learning, ADASYN helps reduce bias and adaptively shift the classification decision boundary towards difficult examples. This approach effectively improves learning outcomes across various machine learning data sets, making it a valuable tool for tackling imbalanced data sets in modern data mining applications.

Temraz and Keane proposed a data augmentation method that generates synthetic, counterfactual instances in the minority class. Unlike other oversampling techniques that interpolate values between instances, this method adaptively combines existing instances from the dataset using actual feature values. To generate synthetic instances, the paper deploys a case-based counterfactual method. Counterfactual

methods are developed to generate posthoc examples to explain the predictions of black-box ML models and provide algorithmic recourse for end-users trying to mitigate automated decisions [45]. Hulse *et al.* analyzed eleven learning algorithms on thirty-five real-world datasets to guide machine learning practitioners and suggest future research directions on building classifiers from imbalanced data. This study is unique as no other related work has analyzed class imbalance on such a wide scope [48].

The data augmentation field is vast, and it is especially used in the field of images. Image data augmentation involves creating new images from existing ones by making small adjustments, such as changing brightness, rotating the image, or shifting the subject horizontally or vertically. This technique effectively increases a dataset's size and improves a machine-learning model's robustness. When a model performs differently on training data versus testing data, it's called generalizability. Overfitting occurs when a model has poor generalizability due to being overly trained on the training data. Simple transformations like horizontal flipping, color space augmentations, and random cropping were the earliest demonstrations of the effectiveness of Data augmentation. These transformations address invariances that pose challenges to image recognition tasks. The efficiency of geometric and photometric (color space) conversions was examined in comparative research by Taylor and Nitschke [44]. We looked at geometric changes, including flipping, 0° to 360° rotations and cropping, as well as color space transformations like edge improvement, PCA, and color jittering (random color manipulation). Eight thousand four hundred twenty-one photos with a size of 256×256 from the Caltech101 dataset were used in the 4-fold cross-validation test of the augmentations.

Generative modeling, nicknamed Generative Adversarial Network (GAN), is a fascinating data augmentation method. Generative modeling is constructing artificial instances from a dataset while maintaining the original set's features. The highly intriguing and enormously well-liked generative modeling framework known as GANs results from the above-mentioned adversarial training ideas. GANs are a means to "unlock" more information from a dataset, according to Bowles *et al.* [7].

3.2. Unification in survey data

After augmentation, another perspective is the unification. The challenges of unification rather than its benefits, particularly concerning long-term economic growth and the practical aspects of societal and political integration. The extent to which the vocabulary and understanding of unification are unknown is still uncertain [37]. There are various types of categorical data, such as text data, DNA sequences, and Census Bureau data, that humans easily understand. Still, many classification systems, like support vector machines (SVM), require numerical data representations. Most learning techniques transform categorical data into binary values to handle this, which can result in high dimensionality and sparsity.

CNFL uses eigen-decomposition to convert the proximity matrix into a reduced space that can be used for classification or clustering. It first employs simple matching

to measure the closeness between instances [21]. Mamabolo and Myres provided two significant contributions. Firstly, it outlines a precise and reproducible 8-step process for questionnaire development utilizing qualitative research, which enhances the methodology for mixed-method designs. Secondly, the study creates a research tool for measuring the extent of entrepreneurial skills. Ultimately, the findings offer implications for research methodology, entrepreneurship scholarships, and practical applications [32, 49]. In data analysis, it is expected to ask meaningless questions.

Understanding data scaling can sometimes help us identify nonsense, but we must use proper logic. Giordan and Diana developed a new clustering technique that addresses two common cluster analysis issues: group size selection and scale invariance. The method employs a multinomial model, a cluster tree, and a pruning approach to group objects. Two types of pruning are examined using simulations [20]. When dealing with real-world problems, data may include numeric and categorical variables. While many regression algorithms work well with numeric variables, categorical variables require additional considerations. However, decision tree algorithms can estimate targets based on specified rules and handle categorical and numeric variables. Kim and Hong proposed a new hybrid model combining a decision tree with another regression algorithm to analyze mixed data. The algorithm was evaluated on twelve datasets and achieved better or comparable accuracy to other methods without significantly increasing computational complexity [25–27].

The decision tree algorithm can handle categorical and numerical variables by evaluating the target based on predefined rules. This feature is used to create a new hybrid model that combines a decision tree with a different regression technique to analyze mixed data. The GA algorithm optimizes the new cost function and produces accurate clustering results. We can evaluate whether a GA-based clustering algorithm suits high-dimensional data collections with mixed features [36]. A novel distance metric is proposed to preserve the order link between ordinal values while measuring the intra-attribute distances of nominal and ordinal characteristics in a unified manner. An entropy-based distance metric for ordinal attributes is devised to estimate the distance between categories of an ordinal attribute, which utilizes the underlying order information. The next step is to generalize this distance measure and suggest a single one that applies to ordinal and nominal attribute categorical data [50].

3.3. Other techniques in survey data

Inference from sample surveys has traditionally focused on functions such as averages and totals of the findings made for the population's participants. However, in scientific applications, the superpopulation parameters linked to a stochastic mechanism assumed to produce the population's observations are frequently of more interest than the finite-population parameters. Even with a modest sampling proportion of the final units, cluster sampling, and conventional design-based variance calculations can significantly underestimate super-population variability [22]. In many empirical applications, there is a chance that mistakes may be associated with clusters. Thus,

it is crucial to strive for accurate statistical inference. We must make sure that our inference takes this into account. Usually, using conventional cluster-robust variance estimators is simple, but things may get complicated occasionally. The two main challenges are dealing with a small number of clusters and figuring out how to define the clusters [11]. Therefore cluster inferencing becomes a more crucial part of survey data analysis. Mixed datasets are frequently subjected to clustering to identify patterns and collect related objects for additional examination. However, it might be not easy to directly apply mathematical operations, such as summing or averaging, to the feature values of these datasets, making clustering mixed data tricky [3].

4. The proposed methodology

Multiple data sources may have different attributes when survey results are gathered. They could be nominal, numeric, or ordinal. Data of all kinds affect survey research. Our conclusions will be more reliable and useful if we incorporate all available facts. One more aspect is there while collecting the data. The number of responses may be small compared to getting a better result with more respondents. In this section, we proposed a model to conquer these deficiencies. The proposed design workflow of this model is given in Figure 2. In the following Subsections, we describe each process of this workflow.

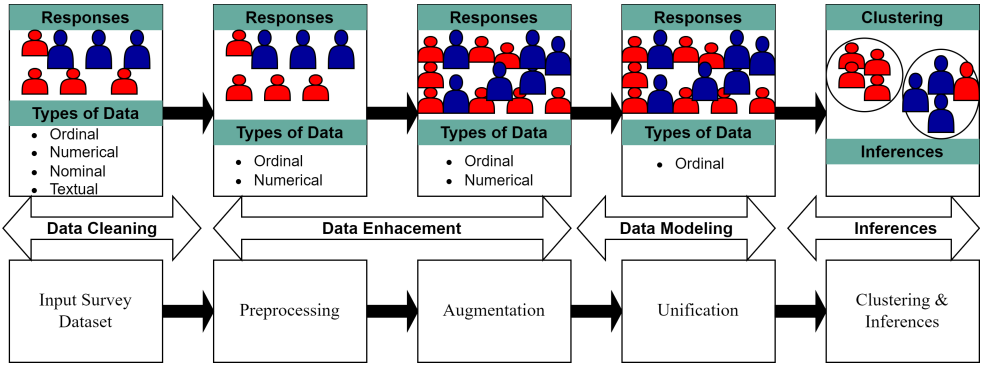


Figure 2. Workflow of the proposed methodology

4.1. Preprocessing

Survey data is essential for preparing the dataset for in-depth analysis and modeling. The reliability and validity of research findings may be increased by resolving difficulties and conflicts for improved data quality, standardization, and representativeness resulting in insightful findings that support well-informed decision-making.

Preprocessing survey data is a vital and complex phase that aims to ensure the gathered data is precise, consistent, and prepared for insightful analysis. Data cleansing is when possible mistakes and missing values are found and dealt with

properly. If attribute values are lacking during the process, the median value based on domain knowledge will fill any gaps. In this article, we focus on two different categories of data, numerical and ordinal, among many others. So the first preprocessing step in this scenario is taking the ordinal and numerical data from the surveyed dataset. After preprocessing and cleaning, we get the dataset for further use. We named it the original dataset.

4.2. Augmentation

We have mentioned SMOTE and ADASYN in the related work section. These two are well-known techniques for data augmentation. These techniques use the nearest neighbor for class imbalance problems with at least two classes. In this article, we do not deal having class imbalance problems. So in this part, another augmentation, a machine learning approach, is used to add more statistical techniques to the already-existing data to expand the diversity of the data. This enhances the generalization and effectiveness of the model. Through augmentation, we attempt to achieve the quality of survey data for a larger population. The Data AUGmentation (DAUG) Algorithm (Algorithm 1) is the pseudo-code for augmentation.

Algorithm 1 *DAUG (S,m,n,P)*

Input: Dataset S, Number of rows m, Number of Columns for using deviation l
Output: Augmented Dataset P

dataset $S[\,]$: Select the numerical and ordinal attributes
n : size of S
m : Number of rows randomly selected from n for augmentation
if $m > n$ **then**
    Reduce the size of m
end if
$m_1[\,]$: make a sample copy of m rows
$m_2[\,]$: make a sample of m rows with random ordinal data distribution
m : Concatenate $m, m_1 \& m_2$
l : Number of columns for considering deviation
for i to l **do**
    Column_medium[i] : choose the medium from each column
    deviation[i] : the deviation for the selected medium from the respected column in each column
    calculate average_deviation[i]
end for
k : Number of rows to add based on average deviation and median
$m = n \times k$
add m rows to dataset P
Normalize P
return P

By applying the DAUG (Algorithm 1), the survey dataset is expended. The original dataset is passed to the model. In the first step, the attributes with numerical and ordinal values are selected and treated as the original dataset. Select the number of rows randomly from the original dataset suitably. Then make two copies of the selected raw data, one for replication and another for different data that have changed ordinal values. Then combine these copies for the augmentation process based on the row-wise mean and standard deviation.

4.3. Unification for data modeling (UFDM)

After augmentation, we consider two types of data values, numerical and ordinal. We make an effort to incorporate survey data that is numerical and ordinal. We use a Gaussian distribution to represent the data. Therefore, we first transform the numerical data into ordinal data that follows the distribution. This process is called a unification for data modeling. The UFDM for unification is given in Algorithm 2.

Algorithm 2 *UFDM (S, a_n)*

Input: Dataset S, Numerical attribute a_n
Output: Unified Dataset D

Select a_n from S
min_a_n : Minimum of the numerical attribute
max_a_n : Maximum of the numerical attribute
avg_a_n : Average of the numerical attribute
$temp\_count[\,]$: for number of occurrence of each number
for i to each number in range min_a_n to max_a_n **do**
    $temp\_count[\,i]$
end for
for i to each row in a_n **do**
    num := row.num
    $temp\_count[\,i] := temp\_count[\,i] + 1$
end for
if $temp\_count[\,]$ is left skewed **then**
    Assignment of ordinal values with making bin following the increasing bin-size from left to right
else if $temp\_count[\,]$ is right skewed **then**
    Assignment of ordinal values with making bin following the decreasing bin size from left to right
else
    Assignment of ordinal values with making bin following the equal bin size from left to right
end if
Replace numerical values with ordinal values and update the dataset with named D
return D

Applying the algorithm UFDM (Algorithm 2), the numerical values are converted to ordinal values through the unification process in the model. The generated dataset from the augmentation process is the input for the unification process. Find the statistics of this dataset, like the minimum, maximum, and average of each numerical attribute. We want to convert numerical data to ordinal data. Then find and count the number of occurrences of each element in ascending order of minimum to maximum values. Adjust the bin size of the ordinal valued bins (based on the Likert scale) accordingly for the skewness nature of the dataset.

4.4. Clustering and inferencing

After augmentation and unification for data modeling, we compare the efficiency of groups through clustering. Clustering entails grouping instances into clusters based on similarity to discover underlying patterns or structures within the dataset. K-means algorithm seeks to optimize the cluster allocations by minimizing the sum of squared distances between data points and their associated centroids. So we use K-means clustering for the whole process. We apply K-means at three levels at the original dataset, after augmentation, and after unification. The clusters made during the process should have improved quality for inferencing so that generalized features can be stated. The governing generalized feature selection is the main focus of inferencing.

4.5. Complexity analysis

The workflow of the proposed technique encapsulates three techniques, namely, Augmentation, Unification, and Clustering. In this subsection, we estimate the computational complexity of the proposed methodology. Let n be the number of rows in the original dataset S.

1. **Augmentation (DAUG):** The steps of the augmentation algorithm (DAUG) are listed in Algorithm 1. The standard deviation in the associated columns is taken into consideration for selecting rows for augmentation Algorithm lines 6–14 are used to calculate each attribute's computation. The time complexity for selecting m rows for augmentation from the dataset S is $O(m)$. The number of columns for considering deviation is l. These columns with selected rows are augmented, therefore, the complexity for this process is $O(m * l)$. The last step is appending the number of k rows, the complexity is $O(k)$. Therefore, the complexity of DAUG Algorithm (Algorithm 1) is $O(m + m * l + k) \approx O(n^2)$.
2. **Unification (UFDM):** The next stage is the unification work: the UFDM Algorithm 2. In UFDM, lines 2–12 are for the unification, and lines 13–20 are for the assignment. Let a_n be the number of numerical attributes. The range of numerical values is in r. The complexity for finding the minimum, maximum, and average of each attribute is $O(a_n * n)$. The complexity for fitting the ordinal values according to the range of numerical values is $O(r * n)$. The last step to replacing the numerical values with corresponding ordinal values

is $O(n * 1)$ complexity.Therefore, the complexity of UFDM Algorithm (Algorithm 2) is $O(a_n + r * n + n) \approx O(n^2)$.

3. **Clustering:** Let the number of desired clusters be (t), the number of rows to be clustered be (n), and the number of iterations until convergence be given by (i). The number of the attributes (a_n) determines the complexity of the K-means method. The Clustering is of $O(t * n * i * a_n) \approx O(n^3)$ [1, 29, 35, 51].

K-means is susceptible to the presence of outliers and is known to perform poorly in the presence of outliers. However, there are several other clustering algorithms, e.g., DBSCAN, hierarchical clustering [18], etc. that handle outliers at the cost of higher complexity [29]. However, this is the future direction of this work.

5. Experimental results and analysis

In this section, we experiment with two datasets and use them to illustrate the proposed methodology and select the generalized features. The performance of clustering algorithms can be assessed using a wide range of metrics, which are utilized depending on a particular task and objectives of the clustering method. In this work, we have considered three performance metric measures: the Silhouette scores [39], Calinski Harabasz Index [10], and Silhouette Analysis plot [39].

Silhouette scores. The silhouette score calculates how well each data point fits into its allocated cluster. This is calculated as the ratio of the mean distance between a data point and all the remaining data points in a comparable cluster to the average distance between a data point and all similar data points in the closest cluster. A higher silhouette score means that the data points have been successfully divided into different clusters that are uniform inside and well-separated by the clustering method.

Calinski Harabasz index. On the contrary, a higher score on the Calinski-Harabasz index denotes superior clustering efficiency. It evaluates the ratio of around-cluster variation to within-cluster variance, which implies how well the Calinski-Harabasz index consider both the gap between clusters and the compactness of each cluster.

Silhouette analysis plot. Each data point's silhouette scores are displayed on the silhouette analysis plot, showing the way each one fits into the cluster to which it was assigned. The range of a silhouette score is from -1 to 1: A clustering allocation with an average of $+1$ is considered successful, whereas one with a value of 0 is considered unclear. A good clustering solution has most data points near $+1$, denoting clearly defined clusters, whereas a not-good clustering solution has values close to 0 or negative values, signifying overlaps or incorrect assignments. By finding the clusters with the greatest average silhouette score representing the most distinct and well-separated, the plot aids in determining the ideal number of clusters. It sheds light on how the quality of clustering and cluster numbers are traded off.

Experimental setup. We have used the proposed model and the clustering technique in the Anaconda edition of Python 3.7 on Windows 10 PC with an Intel Core $i5$ CPU ($2.0GHz$) and $4GB$ of RAM and 64-bit operating system, x64-based processor. In addition to *sklearn*, *metplotlib*, the *pandas* are also used for reading data and visualizing it graphically. We enhanced a Python module of our model to allow for simple code implication. The experiment was conducted within Jupyter Notebooks, using its open-source libraries to speed up and simplify the development process.

Datasets. For our experiment, we have taken two benchmark datasets that are freely available. These datasets are collected from the surveys. These datasets can be downloaded from *Kaggle*, a website with modeling and analysis competitions where data miners compete to create the most effective models using data posted by businesses, researchers, and other users. The following datasets are taken:

- Dataset I: Entrepreneurial Competency Survey.
- Dataset II: Breast Cancer Survey.

5.1. Dataset I: entrepreneurial competency survey

We have collected a dataset [28] to accomplish insightful information about the connection between university students' entrepreneurial habits. This survey aimed to gather data for the students' entrepreneurial propensities levels. Two hundred nineteen responses from survey respondents who were university students make up the dataset we used for this study. Different abbreviations are used for the dataset. These are briefly listed in Table 2.

Table 2
Abbreviation used for Education Sector and Features

Education Sector	Abbr.	Features	Abbr.
Art, Music or Design	AMD	Age	A1
Economic Sciences, Business Studies, Commerce and Law	ESBSCL	Perseverance	A2
Engineering Sciences	EC	DesireToTakeInitiative	A3
Humanities and Social Sciences	HSS	Competitiveness	A4
Language and Cultural Studies	LCS	SelfReliance	A5
Mathematics or Natural Sciences	MNS	StrongNeedToAchieve	A6
Medicine, Health Sciences	MHS	SelfConfidence	A7
Others	OT	GoodPhysicalHealth	A8
Teaching Degree (e.g., B.Ed)	TD		

5.1.1. Dataset description

This dataset, which has two hundred nineteen instances, comprises nine features in the form of attributes, i.e., Age (A1), Perseverance (A2), DesireToTakeInitiative (A3), Competitiveness (A4), SelfReliance (A4), StrongNeedToAchieve (A6),

SelfConfidence (A7), GoodPhysicalHealth (A8), and EducationSector. Age is the numerical data. Perseverance, DesireToTakeInitiative, Competitiveness, SelfReliance, StrongNeedToAchieve, SelfConfidence, and GoodPhysicalHealth are in ordinal data. EducationSector is categorical data.

5.1.2. Statistical analysis

All these features and their overall and attribute-wise mean and standard deviations received from the survey are given in Table 3.

Table 3
Overall and Attribute-wise mean and standard deviations of Original Survey Data

Education Sector		Attributes							
		A1	A2	A3	A4	A5	A6	A7	A8
AMD	Mean	20.33	3.19	3.38	3.43	3.57	3.76	3.67	3.38
	StdDev	1.21	1.01	1.40	1.22	1.14	1.23	1.17	1.25
ESBSCL	Mean	19.56	3.38	3.72	3.47	3.75	4.09	3.56	3.63
	StdDev	1.64	0.96	1.04	1.09	0.94	1.04	1.09	1.32
ES	Mean	19.74	3.38	3.72	3.72	3.81	4.02	3.62	3.61
	StdDev	1.23	1.01	1.02	1.02	0.98	0.90	1.10	0.99
HSS	Mean	19.60	3.40	3.60	3.00	4.00	3.80	3.60	3.60
	StdDev	0.80	0.80	1.02	1.41	1.10	0.98	1.02	1.02
LCS	Mean	19.00	3.00	5.00	3.00	3.00	5.00	5.00	2.00
	StdDev	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MNS	Mean	18.75	3.00	3.25	3.25	2.25	3.00	3.25	3.75
	StdDev	1.17	0.89	1.50	0.98	1.20	1.21	0.81	1.21
MHS	Mean	19.60	3.40	3.20	3.40	3.90	3.60	3.50	3.70
	StdDev	1.20	1.20	1.66	1.56	1.22	1.36	1.28	1.27
OT	Mean	20.00	3.25	3.35	3.45	3.45	3.35	3.30	3.30
	StdDev	0.95	0.83	1.28	1.02	1.07	0.96	1.05	0.95
TD	Mean	19.00	4.00	3.67	3.67	3.67	4.00	3.33	3.67
	StdDev	0.82	0.82	1.25	1.25	1.25	0.82	1.70	1.25
Overall	Mean	19.75	3.35	3.62	3.59	3.72	3.91	3.58	3.56
	StdDev	1.29	0.99	1.15	1.11	1.05	1.02	1.12	1.10

Survey data is augmented with the help of the proposed data augmentation techniques (Algorithm 1) to increase the size of the dataset. Table 4 shows the augmented dataset's overall and attribute-wise mean and standard deviations, with 1676 instances.

Table 4
Overall and Attribute-wise mean and standard deviations of Augmented Survey Data

Education Sector		Attributes							
		A1	A2	A3	A4	A5	A6	A7	**A8**
AMD	Mean	20.48	3.23	2.70	3.11	3.43	3.89	3.95	3.48
	StdDev	1.31	1.04	1.39	1.25	1.07	1.05	1.21	1.31
ESBSCL	Mean	19.53	3.27	3.01	3.05	3.56	4.07	3.93	3.55
	StdDev	1.55	0.90	1.37	1.10	0.87	0.91	1.14	1.29
ES	Mean	19.70	3.41	2.89	3.36	3.65	4.06	3.98	3.65
	StdDev	1.18	0.98	1.30	1.15	0.94	0.76	1.08	0.96
HSS	Mean	19.73	3.45	2.95	2.64	3.64	3.73	3.82	3.64
	StdDev	0.86	0.78	1.30	1.23	1.07	0.86	1.11	0.98
LCS	Mean	19.00	3.00	3.50	2.67	3.00	4.67	5.00	2.00
	StdDev	0.00	0.00	1.19	0.47	0.00	0.47	0.00	0.00
MNS	Mean	18.50	3.00	3.25	3.00	2.40	3.20	3.60	3.80
	StdDev	1.28	1.00	1.24	1.10	1.20	0.98	0.80	0.98
MHS	Mean	19.50	3.44	3.11	3.06	3.78	3.56	3.67	3.72
	StdDev	1.38	1.12	1.33	1.47	1.08	1.17	1.20	1.15
OT	Mean	20.08	3.37	2.99	3.08	3.42	3.61	3.87	3.42
	StdDev	1.01	0.78	1.31	1.06	0.91	0.81	1.10	0.94
TD	Mean	18.71	4.00	2.71	3.14	3.43	4.00	3.71	3.57
	StdDev	0.70	0.93	1.46	1.36	1.18	0.76	1.75	1.40
Overall	Mean	19.74	3.37	2.92	3.22	3.56	3.96	3.94	3.59
	StdDev	1.28	0.96	1.32	1.18	0.99	0.88	1.12	1.07

Error bar line graphs. Error bar line graphs are used to visualize and analyze data and provide essential insights into a dataset's consistency and variability. The distribution of the data around the mean value is revealed by these graphical representations, which aid in determining the relevance of the gathered data. The size of the error bar line graphs, which are frequently represented by standard deviation, effectively conveys how far a given data point deviates from the mean. A small standard deviation bar denotes minimal variability and a higher degree of confidence in the correctness of the data. It also indicates that the data points are closely grouped around the mean. On the other hand, a bigger standard deviation bar highlights greater variability and maybe more uncertainty by showing a wider range of data points away from the mean.

The comparison for the original and augmented data error bar line graphs is shown in Figure 3. This shows the attribute-wise comparison. The blue lines are for the original dataset, and the red lines are for the augmented dataset. In most attributes, the overlapping area shows that the augmented dataset does not deviate from the original data. This process can access the augmented dataset as a large population. And the inferences from the augmentation process we get are the more generalized inferences to make decisions.

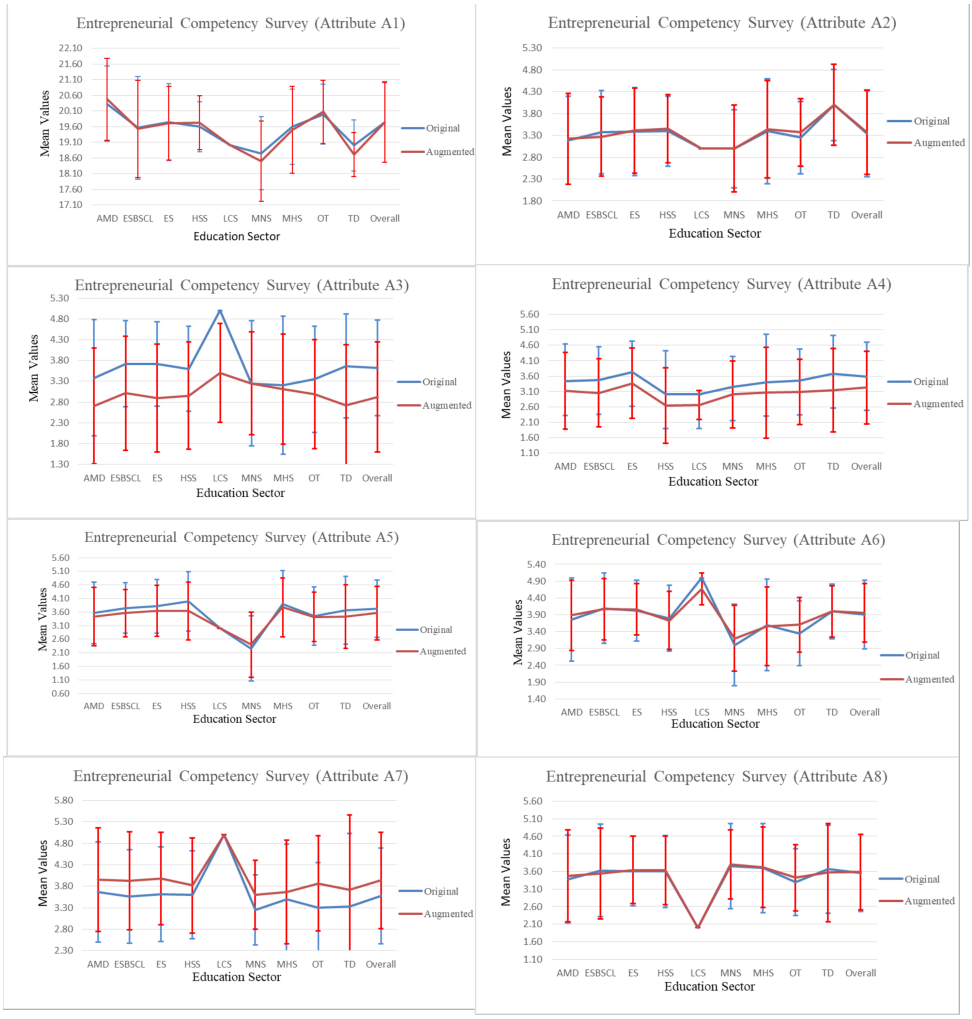


Figure 3. Error bar line graphs for Entrepreneurial Competency survey

5.1.3. Clustering results

We explore our experiment in three parts. In the first part, we considered the original dataset. First, we preprocessed the dataset and separated numerical and ordinal type attributes from the dataset. Then we apply the K-means algorithm for two to nine clusters on the original dataset collected from the survey.

As we have only two hundred nineteen instances in the dataset. So in the second part, we apply the data augmentation techniques to have a proper number of instances. In this manner, we will have sufficient instances and expect to get better inferences through clustering. In the third part, we convert the numerical data to the ordinal data according to Gaussian distribution. After unification, we again apply the K-means algorithm to find the clustering behavior.

In Figures 4, 5, and 6, we have a Silhouette Analysis plot on original, augmented, and unified datasets, respectively. The measurements are shown on two to nine clusters using the K-means algorithm. The resulting plot displays the mean silhouette score for every single clustering solution and the silhouette scores for each cluster data point. High silhouette scores for every point of data and an elevated average silhouette score are desirable as they demonstrate that the data points are correctly segregated and clustered.

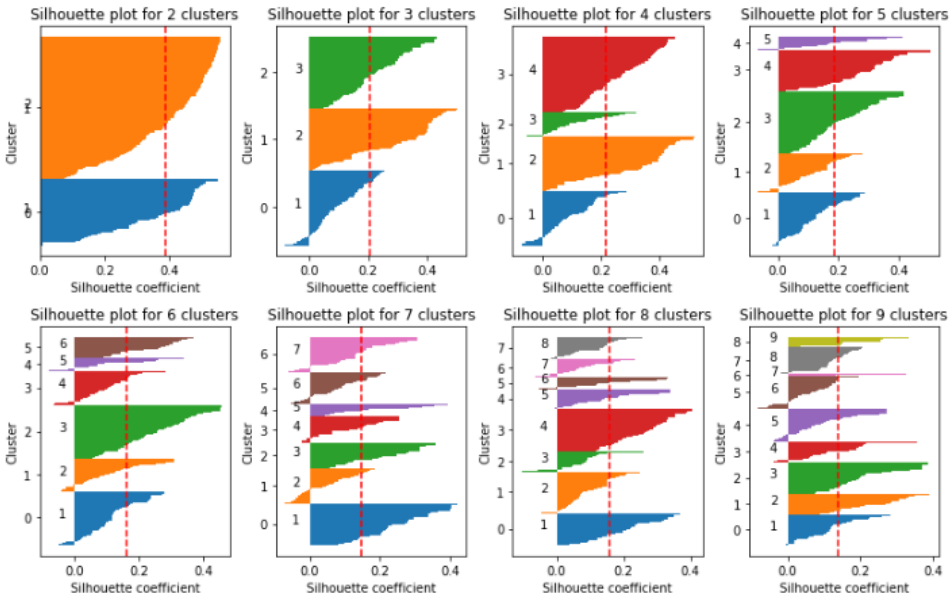


Figure 4. Silhouette analysis plots for original dataset clustering

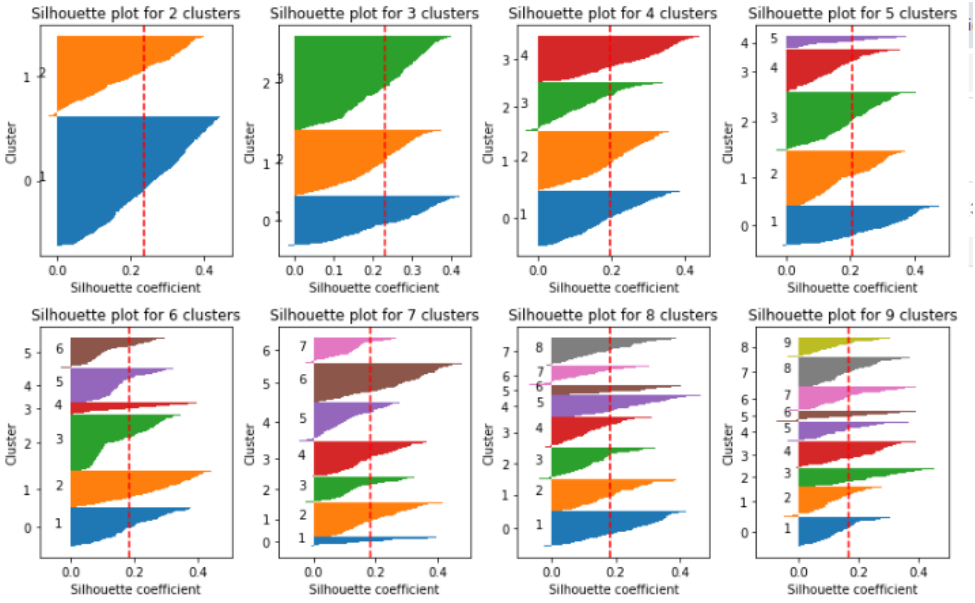


Figure 5. Silhouette analysis plots for augmented dataset clustering

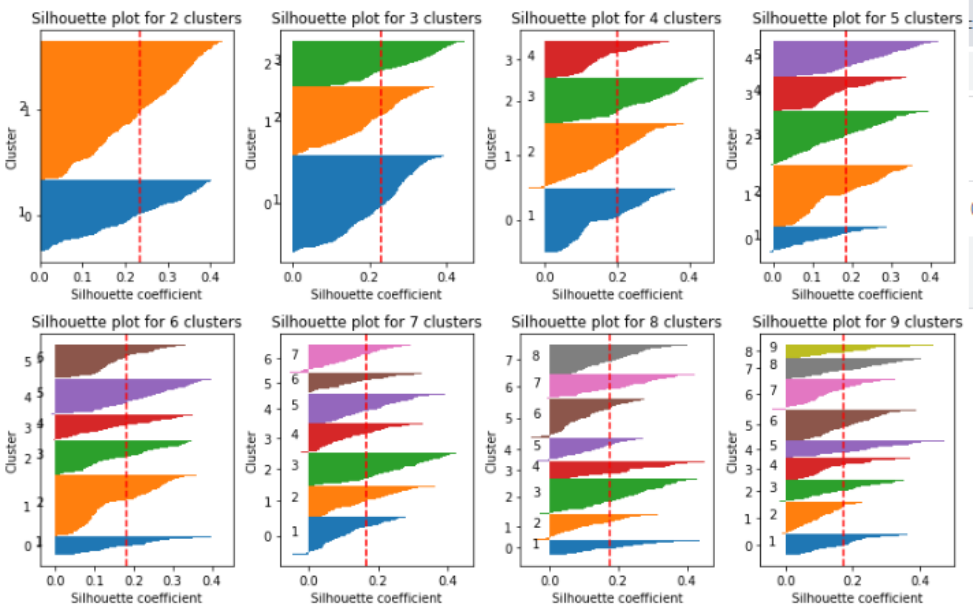


Figure 6. Silhouette analysis plots for unified dataset clustering

5.1.4. Result analysis

In Figure 7, the average Silhouette scores and Calinski-Harabasz indices are shown for the three datasets: one is the original, the second dataset is the one after the augmentation dataset, and the third one is the dataset after unification on three to nine clusters; these scores suggest that higher scores at the same number of the cluster have more data points that were successfully divided into different clusters which are similar inside and distinct from one another.

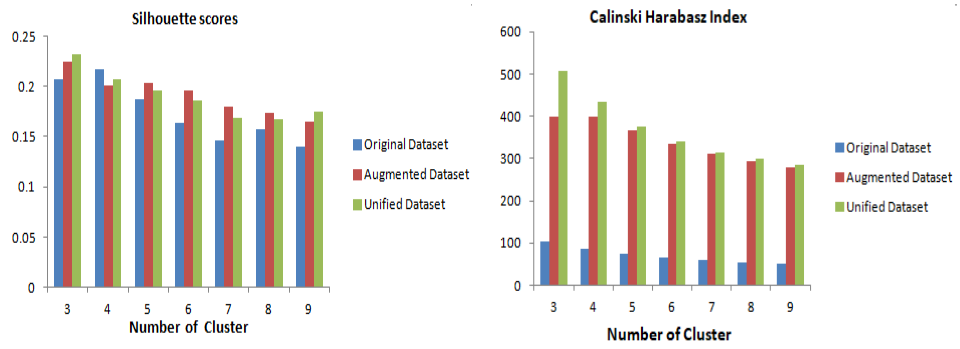


Figure 7. Performance metrics

For example, in Figure 7, at cluster nine, we have Silhouette scores of 0.1397 for the original dataset, 0.1651 for the Augmented dataset, and 0.1746 for the Unified dataset. This means efficiency is improved in clustering after augmentation and unification. The Calinski-Harabasz index considers both the distance between clusters and the compactness of each cluster. The index is higher at each cluster after, one by one, augmentation and unification. Figures 4, 5, and 6 are the complete measurement of Silhouette scores for each data point in each cluster, as well as the average silhouette score for the entire clustering of original, augmented, and unified dataset.

These outcomes show that the clustering procedure has successfully assigned each response to the cluster most closely resembling its features. It produces well-defined clusters with internally comparable replies and clear distinctions between other groups. So, the inferences from these datasets are shown in Table 5. We collected more generalized inferences by ML technique, augmentation.

Table 5
Selected Features for Entrepreneurial Competency

Dataset	Inferences (Competency Factors)
Original Dataset	StongNeedtoAchieve, Desireto TakeInitiative
Augmented Dataset	StongNeedtoAchieve, DesireTo TakeInitiative, Self Confidence
Unified Dataset	StongNeedtoAchieve, DesireTo TakeInitiative, Self Confidence

5.1.5. Runtime analysis

As shown in Figure 8, the state-of-the-art, SOM takes a longer time to train, particularly for big and high-dimensional datasets. It depends on the variables like the amount of data, the network, and the quantity of training iterations [4]. The size and density of the dataset affect how long DBSCAN takes to run. It may be less effective on large datasets, but it works well on datasets with different cluster densities [31]. That is why, we got higher runtime in our dataset available in Figure 8. Our methodology with K-means: among the three, the K-means is frequently the quickest. The convergence speed, which is determined by the start centroids and data distribution, might, however, affect the actual time.

We assess the runtime of the Entrepreneurial Competency dataset in the following Figure 8 for these three techniques at various cluster counts. We used the running time in seconds for SOM and our methodology while we used a logarithmic scale of time for DBSCAN techniques. It can be seen that the proposed technique performs better than the SOTA methods.

#Clust	SOTA		Proposed
	SOM	DBSCAN	
3	5.0	41.2	1.3
4	5.2	41.1	1.1
5	5.0	40.9	2.1
6	5.0	41.3	1.3
7	4.9	40.7	1.2
8	5.0	41.6	1.4
9	4.9	41.0	1.3

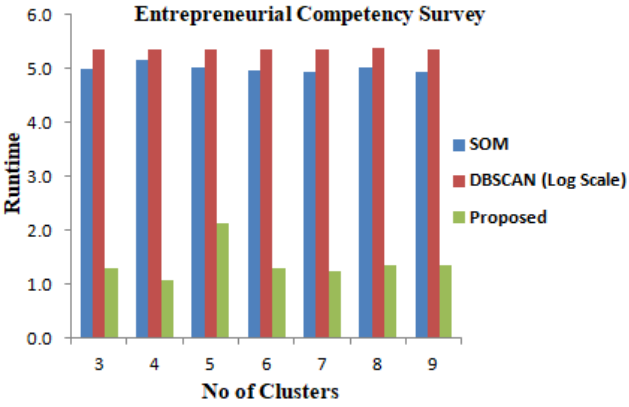


Figure 8. Runtime comparision for Entrepreneurial Competency survey

5.2. Dataset II: breast cancer survey

Next, we have taken a benchmark breast cancer survey dataset for this case study. In this dataset, there are six hundred ninety-nine responses. The dataset values are ordinal. We considered nine features for our study. The dataset, made up of clinical cases Dr. Wolberg documented, is distinguished by the data's arrival time. The dataset attempts to make it easier to forecast the occurrence of breast cancer. An individual code number that serves as an identification for each sample represents it. The features of each sample are then described using a set of Nine attributes. These characteristics include numerical measurements with a range of one to ten.

5.2.1. Dataset description

The survey aimed to gather data on Breast Cancer prediction. The dataset we used for this study comprises six hundred ninety-nine responses from survey respondents. Different abbreviations are used for the dataset. These are briefly described in Table 6. It is compassing attributes such as Clump Thickness (B1), Uniformity of Cell Size (B2), Uniformity of Cell Shape (B3), Marginal Adhesion (B4), Single Epithelial Cell Size (B5), Bare Nuclei (B6), Bland Chromatin (B7), Normal Nucleoli (B8), and Mitoses (B9). In combination, these characteristics capture crucial cell behavior and morphology features that point to probable malignancy. The construction and assessment of breast cancer prediction models are therefore made possible by the extensive set of features with associated diagnostic labels provided by this dataset.

Table 6
Abbreviation used for Breast Cancer survey

Features	Abbr.
Clump Thickness	B1
Uniformity of Cell Size	B2
Uniformity of Cell Shape	B3
Marginal Adhesion	B4
Single Epithelial Cell Size	B5
Bare Nuclei	B6
Bland Chromatin	B7
Normal Nucleoli	B8
Mitoses	B9

5.2.2. Statistical analysis

The mean values and standard deviations of these features in original and after ML techniques, augmented data are given in Table 7.

Table 7
Features and their corresponding mean values and standard deviation

Dataset		Attributes								
		B1	B2	B3	B4	B5	B6	B7	B8	B9
Original data	Mean	4.42	3.13	3.21	2.81	3.22	3.54	3.44	2.87	1.59
	StdDev	2.81	3.05	2.97	2.85	2.21	3.64	2.44	3.05	1.71
Augmented data	Mean	4.02	2.98	3.16	2.95	3.40	3.80	3.84	3.39	2.42
	StdDev	2.35	2.88	2.76	2.73	2.13	3.51	2.55	3.26	2.85

Error line bar graphs. Earlier, we mentioned the error line bar graphs in Subsection 5.1.2. Here we give some information on the Breast Cancer survey dataset. Figure 9 compares the original and augmented data error Line Bar Graphs. The comparison of attributes is also demonstrated here. The red lines represent the augmented dataset, and the blue lines represent the original dataset. The overlapping region for most attributes demonstrates that the enhanced dataset does not diverge from the

original data. This technique allows access to the enormous population of the expanded dataset. The inferences we draw from the augmentation process are those that are more broadly applicable to a significantly large population.

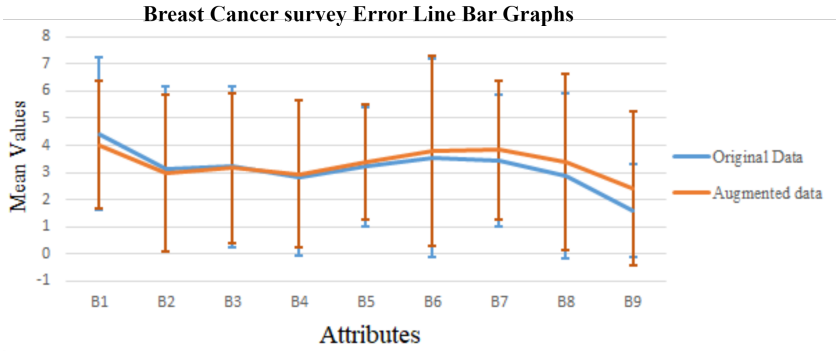


Figure 9. Error line bar graphs for Breast Cancer survey

5.2.3. Clustering results

We break up our investigation into two sections. We use the original dataset in the first section. The dataset was initially preprocessed and then characteristics of the ordinal type were extracted. The original dataset gathered from the survey is then subjected to the K-means method for two to nine clusters. Since the dataset has 699 occurrences only, we use the proposed data augmentation techniques (Algorithm 1) in the second section to have the right number of instances. In this way, we will have enough examples and may use clustering more effectively to draw more accurate conclusions. To learn more about the behavior of the clustering, we employ the K-means technique. The K-means technique displays the measurements on two to nine clusters. The resultant figures show the silhouette scores (see Fig. 10) for each cluster data point and the mean silhouette score for each clustering solution.

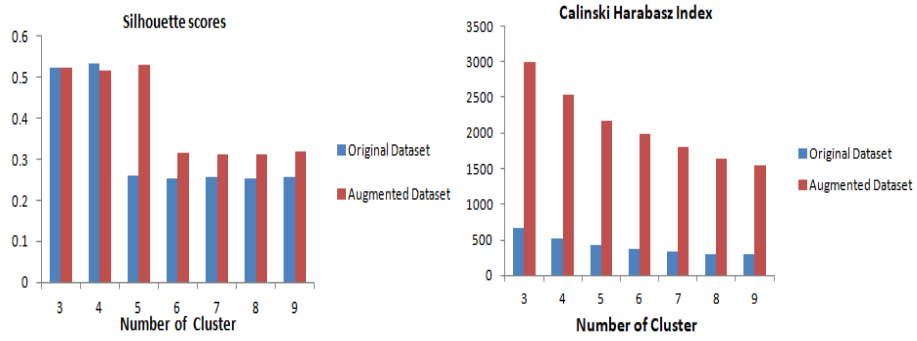


Figure 10. Performance metrics

Every data point should have a high silhouette score, and the average silhouette score should be high since these metrics show that the data points are appropriately grouped and clustered (see Fig. 11, 12).

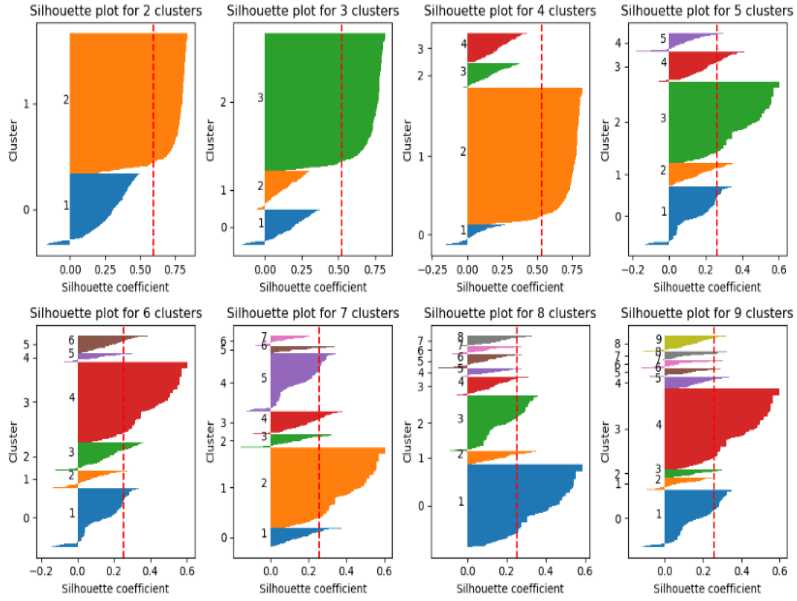


Figure 11. Silhouette analysis plots for clustering of the *original* dataset

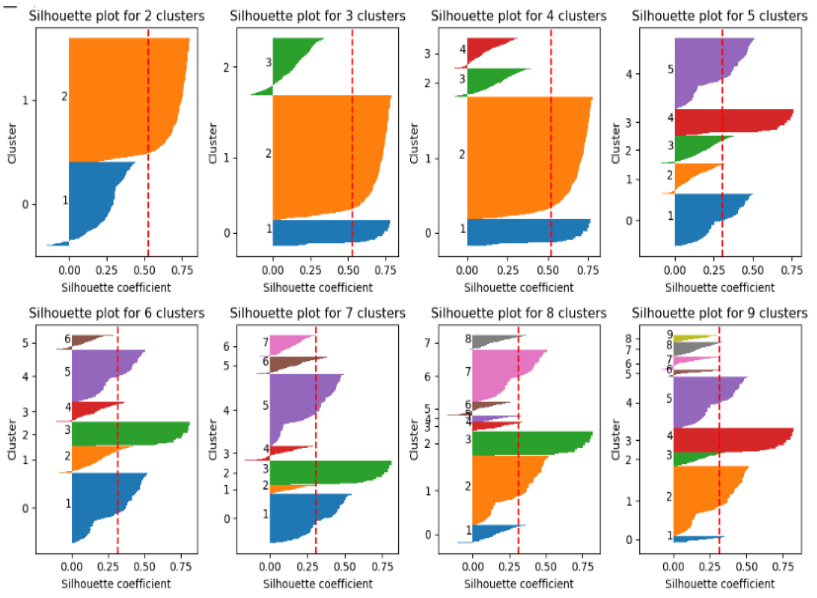


Figure 12. Silhouette analysis plots for clustering of the *augmented* dataset

5.2.4. Result analysis

The results demonstrate that each response was effectively allocated to the cluster that best matched its characteristics. This resulted in well-defined clusters with internally similar responses and noticeable differences between groups. Therefore, the conclusions drawn from these datasets are displayed in Table 8. We utilized the ML approach of augmentation to acquire more generalized inferences.

Table 8
Selected features for the Breast Cancer survey data

Dataset	Inferences (Selected Features)
Original Dataset	Bare Nuclei, Clump Thickness
Augmented Dataset	Bare Nuclei, Clump Thickness, Uniformity cell size, mitoses

5.2.5. Runtime analysis

We have mentioned the runtime analysis for the SOM and DBSCAN in Subsection 5.2.5. In the following Figure 13, we measure the runtime of the Breast Cancer Survey dataset for these three approaches at different cluster counts. For the SOM and the proposed methods, we take runtime in seconds; for DBSCAN techniques, we use the logarithmic scale for time. We may conclude that our strategy outperforms the SOTA techniques.

#Clust	SOTA		Proposed
	SOM	DBSCAN	
3	7.5	97.7	1.0
4	7.6	98.5	1.5
5	7.7	96.9	1.3
6	7.6	97.5	1.3
7	7.7	97.6	1.3
8	7.7	97.7	1.3
9	7.7	97.5	1.2

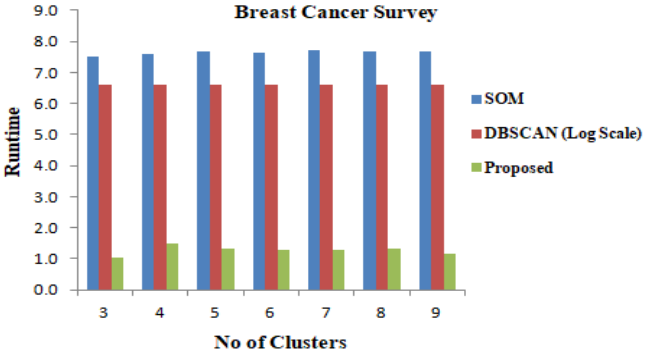


Figure 13. Runtime comparision for Breast Cancer survey

5.3. Discussion

The results obtained with the proposed method suggest that the quality and dependability of inferential findings for a large population from a small population can be improved using augmentation and unification procedures. By extending the existing data with methods such as mean and standard deviation, augmentation improves the dataset's representativeness. Augmentation lowers the chance of bias and improves the generalizability of the inferences made from the analysis. On the other side,

unification refers to combining numerical and ordinal datasets. Unification enables the fusion of many viewpoints and data modalities. Insightful findings, more reliable forecasts, and more precise modeling can result from this.

Scalability. We have proposed three algorithms in this paper. If we consider fewer attributes, the time complexity of the augmentation and unification algorithms is nearer to linear; however, while applying K-means, it may be quadratic complexity and needs to be addressed for scalability. On the other hand, DBSCAN is of quadratic complexity which may also need to be addressed [14]. Since DBSCAN, despite quadratic complexity is made scalable; the proposed algorithms lie in between linear to quadratic and could be made better scalable. This is an area of future work.

In Subsection 4.5, we have discussed computational complexity, which is polynomial between linear and quadratic for both, the augmentation and unification algorithms, and it is very obvious that as the scaling happens the time increases. So, it is crucial to manage the resource demands.

Presence of outliers. We have presented two algorithms, DAUG for data augmentation and UFDM for unification. In both algorithms, the presence of outliers may occur at two stages, one, at the raw data stage, and second, in the outputs of the involved processing techniques.

Raw ordinal data, which is bounded by a few labels, leaves no scope for outliers. However, numerical attributes are prone to outliers, though this could be handled using normalization techniques, such as Z-score. If we have ordinal values we may use the median centering instead of mean [19].

However, in the second stage, the K-means algorithm is prone to outliers. It is well known that K-means performs inappropriately when there are outliers present and is sensitive to their existence. A robust multi-view K-means method with outlier detection to remove the class outliers and attribute outliers can be applied [13]. To inherit the effectiveness of the classical K-means algorithm, with a low time complexity these methods are applied. This is the direction that this work will take going forward.

However, it is crucial to remember that the effectiveness of these strategies depends on choosing the techniques for the appropriateness of the augmentation and unification methods used.

6. Conclusion

In this work, we proposed our approach into two steps and applied the K-means clustering algorithm at each step. We first apply the augmentation technique to generate enough instances to incorporate the richness of data. We measure the deviation of augmented data from original data with descriptive statistical measures. The results are compared for every attribute so that we can employ our method suitable for considering the whole population. The overlapping region for most attributes

demonstrates that the enhanced dataset does not diverge much from the original data. This technique allows access to the huge population of the expanded dataset. The inferences we draw from the augmentation process should also apply to larger survey sizes, however, this is the future direction of the work. Next, we performed the clustering and measured its effectiveness in every aspect. We have used many efficiency metrics for clustering. We included performance metrics like Silhouette's scores, Calinski Harabasz Index, and Silhouette Analysis Plots. The resultant outputs enhanced average and high silhouette scores for each data point show that the data points are appropriately grouped and clustered. Similar to the low Calinski-Harabasz index, the high Calinski-Harabasz index gives well-defined clusters with internally similar responses and apparent distinctions between other groups. It also indicates the distance between clusters and the compactness of each cluster. After augmentation, some numerical attributes may be present in the dataset. So in our next step, we unified the dataset and converted it into the ordinal dataset. This process also helps in generalizing the results of the inferences. After each step, we apply the K-means clustering algorithm and compare the clustering efficiency metrics at different numbers of clusters. Our proposed method shows that efficiency is improved in all such cases. At last, we come to the generalized inferences part. The outcome of both datasets is the selection of the most effective attributes for deciding whether to find the entrepreneurial competency or factors governing breast cancer. Such improved results give better inferences for decision-making. In the future, we would like to use high-dimensional attribute space.

Acknowledgments. The authors thank the reviewer(s) for their insightful comments and suggestions. The authors also express their gratitude to the Editor-in-Chief, the Editor, and the Editorial Office Assistant(s) of this journal for managing this manuscript.

Data and code availability. The program code, data, and artefacts used in this work are publicly available through the *GitHub* repository[1], necessary to run and execute for interpreting, replicating, and building on the findings reported in the paper.

Declaration of competing interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Aggarwal C.C.: An introduction to Cluster Analysis. In: C.C. Aggarwal, C.K. Reddy (eds.), *Data clustering. Algorithms and Applications* chapter 1, pp. 1–28, Chapman and Hall/CRC, 2018. doi: 10.1201/9781315373515-1.

[2] Agrawal T., Choudhary P.: Segmentation and classification on chest radiography: a systematic survey, *The Visual Computer*, vol. 39(3), pp. 875–913, 2023.

[1] https://github.com/Bhuppigithub/Clustering_Inferences_with_Augmentation

[3] Ahmad A., Khan S.S.: Survey of State-of-the-Art Mixed Data Clustering Algorithms, *IEEE Access*, vol. 7, pp. 31883–31902, 2019. doi: 10.1109/access.2019.2903568.

[4] Back B., Sere K., Vanharanta H.: Managing complexity in large databases using self-organizing maps, *Accounting, Management and Information Technologies*, vol. 8(4), pp. 191–210, 1998. doi: 10.1016/s0959-8022(98)00009-5.

[5] Behrend T.S., Sharek D.J., Meade A.W., Wiebe E.N.: The viability of crowdsourcing for survey research, *Behavior Research Methods*, vol. 43, pp. 800–813, 2011. doi: 10.3758/s13428-011-0081-0.

[6] Belloni A., Chernozhukov V., Hansen C.: Inference on Treatment Effects After Selection Amongst High-Dimensional Controls, *The Review Economic Studies*, vol. 81(2), pp. 608–650, 2014. doi: 10.48550/arXiv.1201.0224.

[7] Bowles C., Chen L., Guerrero R., Bentley P., Gunn R., Hammers A., Dickie D.A., Hernández M.V., Wardlaw J., Rueckert D.: GAN augmentation: Augmenting training data using generative adversarial networks, *arXiv: 181010863*, 2018.

[8] Buskirk T.D., Kirchner A., Eck A., Signorino C.S.: An Introduction to Machine Learning Methods for Survey Researchers, *Survey Practice*, vol. 11(1), pp. 1–10, 2018. doi: 10.29115/sp-2018-0004.

[9] Bzdok D., Altman N., Krzywinski M.: Statistics versus machine learning, *Nature Methods*, vol. 15, pp. 233–234, 2018. doi: 10.1038/nmeth.4642.

[10] Caliński T., Harabasz J.: A Dendrite Method for Cluster Analysis, *Communications in Statistics Theory & Methods*, vol. 3(1), pp. 1–27, 1974. doi: 10.1080/03610927408827101.

[11] Cameron A.C., Miller D.L.: A Practitioner's Guide to Cluster-Robust Inference, *Journal Human Resources*, vol. 50(2), pp. 317–372, 2015. doi: 10.3368/jhr.50.2.317.

[12] Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P.: SMOTE: synthetic minority over-sampling technique, *Journal Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002. doi: 10.1613/jair.953.

[13] Chen C., Wang Y., Hu W., Zheng Z.: Robust multi-view K-means clustering with outlier removal, *Knowledge-Based Systems*, vol. 210, 106518, 2020. doi: 10.1016/j.knosys.2020.106518.

[14] Chen Y., Tang S., Bouguila N., Wang C., Du J., Li H.: A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data, *Pattern Recognition*, vol. 83, pp. 375–387, 2018. doi: 10.1016/j.patcog.2018.05.030.

[15] Church A.H., Waclawski J.: *Designing and Using Organizational Surveys: A Seven-Step Process*, John Wiley & Sons, 2001.

[16] Dempster A.P., Laird N.M., Rubin D.B.: Maximum likelihood from incomplete data via the EM algorithm, *Journal Royal Statistical Society: Series B (Methodological)*, vol. 39(1), pp. 1–22, 1977. doi: 10.1111/j.2517-6161.1977.tb01600.x.

[17] van Dyk D.A., Meng X.L.: The Art of Data Augmentation, *Journal of Computational and Graphical Statistics*, vol. 10(1), pp. 1–50, 2001. doi: 10.1198/10618600152418584.

[18] Firdaus S., Uddin M.A.: A survey on clustering algorithms and complexity analysis, *International Journal of Computer Science Issues*, vol. 12(2), 62, 2015.

[19] García-Jara G., Protopapas P., Estévez P.A.: Improving Astronomical Time-series Classification via Data Augmentation with Generative Adversarial Networks, *The Astrophysical Journal*, vol. 935(1), 23, 2022. doi: 10.3847/1538-4357/ac6f5a.

[20] Giordan M., Diana G.: A clustering method for categorical ordinal data, *Communications in Statistics-Theory & Methods*, vol. 40(7), pp. 1315–1334, 2011. doi: 10.1080/03610920903581010.

[21] Golinko E., Sonderman T., Zhu X.: CNFL: Categorical to Numerical Feature Learning for Clustering and Classification. In: *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC), Shenzhen, China*, pp. 585–594, IEEE, 2017. doi: 10.1109/DSC.2017.87.

[22] Graubardand B.I., Korn E.L.: Inference for Superpopulation Parameters using Sample Surveys, *Statistical Science*, vol. 17(1), pp. 73–96, 2002. doi: 10.1214/ss/1023798999.

[23] He H., Bai Y., Garcia E.A., Li S.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong*, pp. 1322–1328, IEEE, 2008. doi: 10.1109/IJCNN.2008.4633969.

[24] Kern C., Klausch T., Kreuter F.: Tree-based machine learning methods for survey research, *Survey Research Methods*, vol. 13 (1), pp. 73–93, 2019.

[25] Kim K., Hong J.S.: A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis, *Pattern Recognition Letters*, vol. 98, pp. 39–45, 2017. doi: 10.1016/j.patrec.2017.08.011.

[26] Kumar B., Kumar R.: Difference-Attribute-Based Clustering for Ordinal Survey Data. In: A.K. Dubey, V. Sugumaran, P.H.J. Chong (eds.), *Advanced IoT Sensors, Networks and Systems. SPIN 2022*, pp. 17–27, Springer, Singapore, 2022. doi: 10.1007/978-981-99-1312-1_2.

[27] Kumar B., Kumar R.: Entropy-based clustering for subspace pattern discovery in ordinal survey data. In: V. Bhateja, X.S. Yang, J. Chun-Wei Lin, R. Das (eds.), *Intelligent Data Engineering and Analytics. FICTA 2022. Smart Innovation, Systems and Technologies*, pp. 509–519, Springer, Singapore, 2022. doi: 10.1007/978-981-19-7524-0_45.

[28] Kumar B., Kumar R.: Unification of Numerical and Ordinal Survey Data for Clustering-based Inferencing, *INFOCOMP Journal Computer Science*, vol. 22(1), 2023. https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/2492.

[29] Kumar R., Rockett P.: Multiobjective genetic algorithm partitioning for hierarchical learning of high-dimensional pattern spaces: a learning-follows-decomposition strategy, *IEEE Transactions on Neural Networks*, vol. 9(5), pp. 822–830, 1998. doi: 10.1109/72.712155.

[30] Ley C., Martin R.K., Pareek A., Groll A., Seil R., Tischer T.: Machine learning and conventional statistics: making sense of the differences, *Knee Surgery, Sports Traumatology, Arthroscopy*, vol. 30(3), pp. 753–757, 2022. doi: 10.1007/s00167-022-06896-6.

[31] Luchi D., Rodrigues A.L., Varejão F.M.: Sampling approaches for applying DBSCAN to large datasets, *Pattern Recognition Letters*, vol. 117, pp. 90–96, 2019. doi: 10.1016/j.patrec.2018.12.010.

[32] Mamabolo M.A., Myres K.: A detailed guide on converting qualitative data into quantitative entrepreneurial skills survey instrument, *The Electronic Journal of Business Research Methods*, vol. 17(3), pp. 102–117, 2019. doi: 10.34190/JBRM.17.3.001.

[33] Mason M.: Sample size and saturation in PhD studies using qualitative interviews, *Forum: Qualitative Sozialforschung/Forum: Qualitative Social Research*, vol. 11(3), 2010. doi: 10.17169/fqs-11.3.1428.

[34] Nardo M.: The quantification of qualitative survey data: a critical assessment, *Journal Economic Surveys*, vol. 17(5), pp. 645–668, 2003. doi: 10.1046/j.1467-6419.2003.00208.x.

[35] Pakhira M.K.: A Linear Time-Complexity k-Means Algorithm Using Cluster Shifting. In: *2014 International Conference on Computational Intelligence and Communication Networks, CICN'2014*, pp. 1047–1051, IEEE, 2014. doi: 10.1109/CICN.2014.220.

[36] Rastogi R., Mondal P., Agarwal K., Gupta R., Jain S.: GA based clustering of mixed data type of attributes (numeric, categorical, ordinal, binary, and ratio-scaled), *BIJIT – BVICAM's International Journal of Information Technology*, vol. 7(2), pp. 861–866, 2015.

[37] Rich T.S.: South Korean perceptions of unification: Evidence from an experimental survey, *Georgetown Journal of International Affairs*, vol. 20, pp. 142–149, 2019. doi: 10.1353/gia.2019.0022.

[38] Rodriguez M.Z., Comin C.H., Casanova D., Bruno O.M., Amancio D.R., Costa L.d.F., Rodrigues F.A.: Clustering algorithms: A comparative approach, *PloS one*, vol. 14(1), e0210236, 2019. doi: 10.1371/journal.pone.0210236.

[39] Rousseeuw P.J.: Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis, *Journal of Computational & Applied Mathematics*, vol. 20, pp. 53–65, 1987. doi: 10.1016/0377-0427(87)90125-7.

[40] Sadh R., Kumar R.: Clustering of Quantitative Survey Data based on Marking Patterns, *INFOCOMP Journal Computer Science*, vol. 19(2), pp. 109–119, 2020.

[41] Sadh R., Kumar R.: Transformation and classification of ordinal survey data, *Computer Science*, vol. 24(2), 2023. doi: 10.7494/csci.2023.24.2.4871.

[42] Schliep E.M., Hoeting J.A.: Data augmentation and parameter expansion for independent or spatially correlated ordinal data, *Computational Statistics & Data Analysis*, vol. 90, pp. 1–14, 2015. doi: 10.1016/j.csda.2015.03.020.

[43] Stevens S.S.: On the theory of scales of measurement, *Science*, vol. 103(2684), pp. 677–680, 1946. doi: 10.1126/science.103.2684.677.

[44] Taylor L., Nitschke G.: Improving Deep Learning with Generic Data Augmentation. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India*, pp. 1542–1547, IEEE, 2018. doi: 10.1109/SSCI.2018.8628742.

[45] Temraz M., Keane M.T.: Solving the class imbalance problem using a counterfactual method for data augmentation, *Machine Learning with Applications*, vol. 9, 100375, 2022. doi: 10.1016/j.mlwa.2022.100375.

[46] Tourangeau R.: Cognitive aspects of survey measurement and mismeasurement, *International Journal of Public Opinion Research*, vol. 15(1), pp. 3–7, 2003. doi: 10.1093/ijpor/15.1.3.

[47] Valsiner J., Molenaar P.C., Lyra M.C.D.P., Chaudhary N.: *Dynamic Process Methodology in the Social and Developmental Sciences*, Springer, New York, 2009.

[48] Van Hulse J., Khoshgoftaar T.M., Napolitano A.: Experimental perspectives on learning from imbalanced data. In: *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pp. 935–942, Association for Computing Machinery, New York, 2007. doi: 10.1145/1273496.1273614.

[49] Velleman P.F., Wilkinson L.: Nominal, ordinal, interval, and ratio typologies are misleading, *The American Statistician*, vol. 47(1), pp. 65–72, 1993. doi: 10.1515/9783110887617.161.

[50] Zhang Y., Cheung Y.M.: Learnable Weighting of Intra-Attribute Distances for Categorical Data Clustering with Nominal and Ordinal Attributes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44(7), pp. 3560–3576, 2021. doi: 10.1109/TPAMI.2021.3056510.

[51] Zhang Y., Cheung Y.M., Tan K.C.: A Unified Entropy-Based Distance Metric for Ordinal-and-Nominal-Attribute Data Clustering, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31(1), pp. 39–52, 2019. doi: 10.1109/TNNLS.2019.2899381.

Affiliations

Bhupendera Kumar
Jawaharlal Nehru University, Data to Knowledge (D2K) Lab, School of Computer & Systems Sciences, New Delhi 110 067, India, bkchauhan86@gmail.com

Rajeev Kumar
Jawaharlal Nehru University, Data to Knowledge (D2K) Lab, School of Computer & Systems Sciences, New Delhi 110 067, India, rajeevkumar.cse@gmail.com

Received: 14.08.2023
Revised: 29.12.2023
Accepted: 08.01.2024

Computer Science • 25(1) 2024 https://doi.org/10.7494/csci.2024.25.1.5330

Jarosław Stańczak

EFFICIENT SELECTION METHODS IN EVOLUTIONARY ALGORITHMS

Abstract *Evolutionary algorithms mimic some elements of the theory of evolution. The survival of individuals and the ability to produce offspring play significant roles in the process of natural evolution. This process is called natural selection. This mechanism is responsible for eliminating weaker members of the population and provides the opportunity for the development of stronger individuals. The evolutionary algorithm, an instance of evolution in the computer environment, also requires a selection method – a computerized version of natural selection. Widely used standard selection methods applied in evolutionary algorithms are usually derived from nature and prefer competition, randomness, and some kind of "fight" among individuals. But the computer environment is quite different from nature. Computer populations of individuals are typically small, making them susceptible to premature convergence towards local extremes. To mitigate this drawback, computer selection methods must incorporate features distinct from those of natural selection. In the computer selection methods randomness, fight, and competition should be controlled or influenced to operate to the desired extent. This work proposes several new methods of individual selection, including various forms of mixed selection, interval selection, and taboo selection. The advantages of incorporating them into the evolutionary algorithm are also demonstrated, using examples based on searching for the maximum α-clique problem and traditional Traveling Salesman Problem (TSP) in comparison with traditionally considered highly efficient tournament selection, deemed ineffective proportional (roulette) selection, and other classical methods.*

Keywords evolutionary algorithms, selection methods, adaptive evolutionary algorithms

Citation Computer Science 25(1) 2024: 95–122

1. Introduction

In the evolution of living creatures, the process of natural selection is the primary force guiding gradual changes in their genomes across consecutive generations and contributing to their development. The process of natural selection is not a single mechanism, it consists of several different elements overlapping each other, with some parts being deterministic and others random. The superposition of these factors affects the success of the individual. The sum of these successes of individuals gives a higher level of development of the total population and species. Hence, the selection can be compared to the strainer filtering out the weakest organisms (in any case the most likely eliminates individuals weaker than the better), thereby enabling the survival of the best features and the accumulation of these traits in subsequent generations.

In a natural environment, it is difficult to say to what extent the content of the genetic code of the individual determines the success of the development. Probably in simple organisms, it matters the most, in higher organisms, which have the ability to learn, the direct influence of their genes is smaller, but of course, the learning ability in some way comes from what is written in the DNA of the organism. Therefore, one can observe the phenomenon in which the future of individuals and species is influenced not only by genetic predisposition but also the skills acquired during individuals' lifetimes. Unfortunately, this feature of living organisms is still weakly used by evolutionary algorithms, although this is certainly a very promising opportunity, developed in memetic algorithms [12] and agent systems. Certainly, the combination of these features in the memetic or evolutionary-agent systems has a great future.

During the early development of genetic or evolutionary algorithms, much attention was devoted to mimicking processes observed in natural evolution [11]. However, applying pure natural selection in computer environment is not possible. Concepts such as strong competition among individuals, randomness, and the fight for survival have given rise to several traditional selection methods used in evolutionary computations (proportional or roulette wheel selection, tournament selection).

However, such an approach, in the case of rather small populations of individuals with which users are dealing in evolutionary algorithms, is often inefficient and leads to their premature convergence to local extremes or do not use the potential inherent in the method of evolution, significantly slowing down the computations. Depending on the method of selection, the algorithm easily falls into either of two disadvantages: the best individuals dominate the entire population and a significant slowdown or blockage of the evolution due to the small diversity of the population (too high selection pressure) or on the other side the exploration of new areas is strong, but promising solutions live too short to find near-optimal solutions (the selection pressure is too weak).

Thus, it can be seen that the selection in the computer environment must operate quite differently than natural selection [17]. Since a universal computer selection

method cannot be found, there is certainly a need to choose a method from existing ones or even develop a method, depending on the specifics of the problem being solved. Furthermore, some researchers consider the selection as one of the genetic operators. Unless there is no doubt that the set of genetic operators should be tailored to the specifics of the solved problem, spreading this idea concerning selection is usually challanging.

Accordingly, the most frequently used are quite typical and basic methods, while it would be possible to obtain much better results using a little more sophisticated methods: adaptive or a method in which the selection is disguised as another mechanism, e.g., the lifetime of an individual [2], etc.

This paper is a continuation of research, presented in [17] with several new selection methods developed from that time. Although there are some theoretical methods, that describe properties of selection methods (Section 2), empirical experiments which show their behavior are also very important (Section 5). Sections 3 and 4 provide detailed descriptions of the discussed selection methods.

2. Properties of selection methods

Selection in evolutionary algorithms is characterized by the concept of selective pressure. It is difficult to strictly define this notion. Mostly, it is described using coefficients, the values of which were estimated for some selection methods. However, these factors never fully reflect the nature of the method. So far, all assessments of the suitability of selection methods have been verified experimentally to observe their performance.

This is due to the absence of an adequate mathematical framework that could explain the theory and methods of operation of evolutionary algorithms, showing how the selection method affects the convergence of the algorithm to the optimum or sub-optimum.

In addition, evolutionary selection is not the only force targeting calculations for more sophisticated evolutionary algorithms. Similar or even greater importance may lie in specialized genetic operators enriched with the knowledge of the task to be solved or adapted from simple methods of local optimization, often used in advanced evolutionary algorithms and memetic algorithms.

Goldberg and Deb [10] proposed a measure of selective pressure called the *takeover time*, representing the number of generations τ needed to fill the entire population of solutions with the best copies of the same individual in the absence of modification by genetic operators (assumed to preserve a copy of the best individual to prevent accidental extinction). Unfortunately, for many of the more complicated selection methods, estimating this time analytically is challenging. Even if it could be calculated, it may be difficult to precisely assess the properties of the particular selection method looking only at its value of τ.

Another coefficient measuring selection characteristics is called *selection intensity* – I [13], defined as follows:

$$\mu_{t+1} = \mu_t + I \cdot \sigma_t \tag{1}$$

where:

μ_t, μ_{t+1} – mean values of the population fitness function values before and after selection,

σ_t – the standard deviation of the population fitness function before selection.

This factor was defined for the theoretical study aiming to assess the convergence of selection methods.

Another method for assessing selection is the concept of *genetic drift*, as presented for instance, in [15]. Genetic drift is a phenomenon observed in evolutionary algorithms, which depends on changing frequencies of genes in the population, consequently leading to the convergence of the population to identical solutions. Genetic drift is defined as follows:

$$r = \frac{E(\sigma')}{\sigma} \tag{2}$$

where:

r – the genetic drift coefficient,

$E(...)$ – a symbol of the expected value,

σ' – the standard deviation of the population fitness function after selection, and σ before selection.

The *selection variance* [5] is a factor easier to determine than the genetic drift in practical computer simulations, describing properties of selection in evolutionary algorithms:

$$V = \frac{\sigma'^2}{\sigma^2} \tag{3}$$

where:

V – the selection variance coefficient,

$E(...)$ – a symbol of the expected value,

σ' – the standard deviation of the population fitness function after selection,

σ – the standard deviation before selection.

The selection of individuals typically results in the *loss of diversity* within the descendant population. This is rather a harmful phenomenon in relatively small computer populations, especially big levels of loss of population diversity. The loss of diversity factor can also serve as a measure of selection properties [5, 6]:

$$p_d = \frac{N - |P(t) \cap P(t+1)|}{N} \tag{4}$$

where:

p_d – the measure of loss of diversity,

N – the cardinality of the population,

$P(t)$, $P(t+1)$ – populations before and after selection.

Loss of diversity in the population as a result of selection action can be minimized using non-standard methods especially designed to obtain that aim. Several of them are presented in the section 3.1. Additionally, a simple parameter indicating the level of the *population diversity* can be introduced. It is the number of different solutions in the population in proportion to the population cardinality – before and after selection. The notion "different solutions" means different in encoded solutions, not only different in the values of the fitness function, because the same values of the fitness function may characterize completely different solutions. The similarity factor in this case is a difference in at least one encoded position[1] (one bit, one city, one graph node, ..., for the real-number problem it can be some vicinity of a given point). Therefore, it is possible to analyze population diversity before and after selection:

$$s_b = \frac{N_{db}}{N_b} \tag{5}$$

$$s_a = \frac{N_{da}}{N_a} \tag{6}$$

where:

s_b, s_a – measures of the population diversity before and after selection,
N_b, N_a – the cardinality of the population before and after selection,
N_{db}, N_{da} – numbers of different solutions (encoding different points in the problem's space) in population before and after selection.

3. Standard selection methods

While numerous selection methods have been invented and can be considered standard, two of them stand out as particularly important: the proportional or roulette wheel method (the first to be used and theoretically analyzed) and tournament selection (known for its excellent practical properties and frequently employed in theoretical research). However, it's worth noting that the remaining methods are also applied in practice and, at times, in theory.

3.1. The proportional or roulette selection

It is one of the oldest selection methods used in genetic algorithms. Every individual in the parent population can have an offspring with a probability proportional to the value of its fitness function. In other words, the probability of selecting each individual is equal to the ratio of the value of its fitness function to the sum of the fitness values for the entire population. Consequently, the probability of selecting the individual and the expected value of descendants for that individual can be expressed using the following formulas:

$$p_l(t+1) = \frac{F_l(t)}{\sum_{j=1}^{\mu+\lambda} F_j(t)} \tag{7}$$

[1]Of course, it is possible to use a stronger similarity criterion with more than one difference in the encoded solution.

$$En_l(t+1) = \frac{\mu * F_l(t)}{\sum_{j=1}^{\mu+\lambda} F_j(t)} \tag{8}$$

where:

$p_l(t+1)$ – the probability of selection of the l-th individual to the descendant population $l \in 1, ..., \mu + \lambda$,

$En_l(t+1)$ – the expected value of descendants of the l-th individual, $l \in 1, ..., \mu + \lambda$,

μ – cardinality of the parent population,

λ – cardinality of the descendant population,

$F_l(t)$, $F_j(t)$ – values of fitness functions for the l-th (j-th) individual.

As it has been stated in some works, this method of selection is proper only for theoretical purposes. The real application of it is rather useless because there are better methods, especially the tournament selection [23].

3.2. The deterministic roulette (or proportional) selection method

The deterministic roulette method is a modified proportional selection in which randomness has been eliminated[2]. Additionally, some scaling skills have been introduced to improve obtaining integer and proper numbers of offspring individuals (the number of offspring should be equal or close to μ).

An individual from the parent population obtains a number of offspring by rounding the ratio of the value of its fitness function to the average value of the fitness function for the entire population to the nearest integer (or integer part), multiplied by the population's cardinality (9). The resulting values are scaled again using a similar ratio to minimize errors in the approximation of offspring cardinality. If there is depletion despite the occurrence, it is populated with the best individuals that have not entered the population or those with the highest discard fractions. The excess is disposed of by removing the appropriate number of the worst-selected individuals.

$$n_l(t+1) = f\left(\frac{\frac{(\mu+\lambda)\cdot F_l(t)}{\sum_{j=1}^{\mu+\lambda} F_j(t)} \cdot \mu}{\sum_{i=1}^{\mu+\lambda} f(\frac{F_i(t)\cdot(\mu+\lambda)}{\sum_{k=1}^{\mu+\lambda} F_k(t)})}\right) \tag{9}$$

where:

$n_l(t+1)$ – the number of offspring of the l-th individual in the descendant population, where $l \in 1, ..., \mu + \lambda$,

$f(...)$ – a function that converts the actual floating point result to integer value,

μ – the cardinality of the parent population,

λ – the cardinality of the descendant population,

$F_l(t)$, $F_j(t)$ – values of fitness functions for the l-th (j-th) individual.

[2]This method is very similar to the *selection by stochastic remainder method with repetitions* where the remaining vacant fractional parts of individuals in the population are supplemented according to the proportional selection.

3.3. The tournament selection

This selection method operates as follows. First, a random collection of a few individuals (greater than one, less than the total population number) is selected, then one of them with the best value of fitness function wins. Individuals are drawn to the tournament among the entire population with equal probability. The most commonly used is a variant in which the two individuals are drawn and to the descendant population the best one is selected, but in the general case, the size of the tournament can be any number (not bigger than the population size), and the best individual from the selected ones is copied to the descendant population. The draw is carried out as many times as there are free places for individuals in the descendant population. The probability of selecting an individual in this method is expressed by formula (10), and the expected number of individuals by formula (11) (based on [3]):

$$p_l(t+1) = \frac{(\mu+\lambda-l+1)^k - (\mu+\lambda-l)^k}{(\mu+\lambda)^k} \tag{10}$$

$$E(n_l(t+1)) = \mu \cdot \frac{(\mu+\lambda-l+1)^k - (\mu+\lambda-l)^k}{(\mu+\lambda)^k} \tag{11}$$

where:

$p_l(t+1)$ – the probability of choosing the l-th individual to the new parent population, where $l \in 1, ..., \mu+\lambda$,
$E(n_l(t+1))$ – the expected number of copies of the l-th solution in the descendant population, where $l \in 1, ..., \mu+\lambda$,
μ – the cardinality of the parent population,
λ – the cardinality of the descendant generation,
k – the size of the tournament.

Contrary to the proportional selection, this method is widely used for both practical and theoretical purposes.

4. Investigated selection methods

4.1. A histogram selection

This method is a result of searching for an algorithm of selection with a high ability to maintain the population diversity while preserving the selective pressure of population growth, as presented in [17]. The version presented below is a slightly modified variant and it is also used as an element of new methods described later in this work.

The first step of this method's operation involves distributing the population based occurring values of the fitness function, hence its similarity to create a histogram and its name. There are two possible variants of performing this step. One ignores solutions with values of fitness function the same as just included in the list. However, quite often it happens that in the population there are also solutions with the same values of the fitness function, but representing completely different solutions.

Ignoring this fact constitutes a significant simplification of the problem and may result in a considerable loss of population diversity. A good solution to this problem requires a comparison of encoded representations of individuals with the same values of fitness functions, and such a mechanism is applied in the considered histogram selection. A proper way to detect such situations remarkably depends on the method of encoding solutions and in many cases can be quite complicated if the encoding method used is ambiguous (several different individuals may encode the same solution). For instance, to solve an instance of the TSP (Traveling Salesman Problem) using EA where a list of visited cities is treated as a solution encoding, several lists may seem not similar at first glance but may encode the same solution, differing the starting point or the travel direction.

Thus, it is necessary to adopt a metric in the domain of coded solutions and accept certain criteria to distinguish solutions. Generally, in the case of binary or integer encoding it can be assumed that the solutions are different if they differ in at least one position. It is possible, however, to assume identical solutions which differ to a greater extent. In the case of real number encoding, it is necessary to apply a certain minimum distance between the solutions below which they are considered to be identical. It would be a factor similar to the crowding factor, widely used in EA.

Regardless of the method used (whether comparing solutions or not), the obtained list of selected individuals is usually shorter than the list of individuals in the parent population ($s \leq \mu + \lambda$ or $s \leq \lambda$), depending on the population development strategy due to possible repetitions of the same solutions in the population.

The second step of histogram selection can also be executed in two ways. The first considered version can be used as an independent selection method and is characterized by a relatively low level of selection pressure. Each individual from the list passes to the offspring population the number of individuals that results from rounded to the nearest integer ratio of its value of fitness function to the average of the fitness function for the list, multiplied by population size. This is illustrated by Formula (12). In this formula, a scaling mechanism is used to minimize rounding errors. If, despite this, the size of the descendant population is greater or less than the expected number of individuals, the population is replenished using the best individuals who have not yet entered the new generation, or the proper number of the weakest among the selected individuals is removed.

$$n_l(t+1) = f\left(s \cdot \frac{F_l(t)}{\sum_{j=1}^{s} F_j(t)} \cdot \frac{\mu}{\sum_{i=1}^{s} f\left(\frac{F_i(t) \cdot s}{\sum_{k=1}^{s} F_k(t)}\right)}\right) \tag{12}$$

where:

$n_l(t+1)$ – the number of offspring, copies of the l-th solution from the list in the descendant population, where $l \in 1, ..., s$,

$f(...)$ – a function that implements one method of converting the real value result to an integer value, it can be either round – which approximates the actual value to the nearest integer or floor – rejecting the fractional part of the calculated number of descendants,

s – the number of different values of the fitness function (or better, different genotypes) distinguished in the resulting list,
$F_l(t)$ – the value of the fitness function for the l-th individual in the current population.

In the second version of histogram selection (flat histogram selection), the survival of the fittest individuals from the prepared list of individuals is performed in this step of the considered method. Thus, the descendant population is more diverse than in the first version of the second step, since it does not contain repeated solutions and even very good individuals are mentioned only once. This selection version has very small selective pressure; it mainly strongly increases the population diversity and rather cannot be used as an independent selection method but only as an auxiliary method or some form of population preprocessing.

4.2. Mixed selection

The histogram selection gives good results in evolutionary computations, preventing the too rapid convergence of the population, but it is characterized by a rather small selection pressure on the population towards promoting the best individuals, much smaller than the deterministic roulette method. It has a long or infinite time of unification of the population. On the contrary, the deterministic roulette method remarkably promotes the best solutions, but this leads to a rapid loss of population diversity and premature convergence. The unification time of the population can be estimated in this method as follows:

$$\tau \leqslant \frac{\ln(\mu)}{\ln(1+\frac{\lambda}{\mu})} \tag{13}$$

where:

τ – time (number of generations or iterations) to fill the population with identical individuals, without affecting the evolutionary operators,
μ – parent population size,
λ – the size of the new generation.

A combination of advantages of both methods, compensating for their shortcomings, can be achieved using a mixed selection. This idea was previously presented in [17]. The mixed selection consists of two component methods with significantly different properties: histogram selection (flat histogram selection or simple histogram selection) which has the property of significantly increasing the diversity of the population, and the deterministic roulette, with a strong focus on promoting the best individuals and thus decreasing the diversity of the population. These methods are randomly selected and executed during the operation of the evolutionary algorithm.

The probability of selection and performance of each method shows the formula (14):

$$\begin{aligned} p_{his}(t+1) &= \begin{cases} p_{his}(t)\cdot(1-a) \text{ for } R(t) > 3\cdot\sigma(F(t)) \\ p_{his}(t)\cdot(1-a)+0.5\cdot a \text{ for } R(t) \geqslant 0.5\cdot\sigma(F(t)) \text{ and } R(t) \leqslant 3\cdot\sigma(F(t) \\ p_{his}(t)\cdot(1-a)+a \text{ for } R(t) < 0.5\cdot\sigma(F(t)) \end{cases} \\ R(t) &= max(F_{av}(t)-F_{min}(t), F_{max}(t)-F_{av}(t)) \\ p_{det} &= 1-p_{his} \end{aligned} \tag{14}$$

where:

$p_{his}(t)$ – the probability of histogram selection,
$p_{det}(t)$ – the probability of selection by deterministic roulette,
$F_{av}(t)$, $F_{min}(t)$, $F_{max}(t)$ – average, minimum, and maximum values of the fitness function in the population,
$\delta(F(t))$ – the standard deviation of the population fitness function.

If the set of values of the fitness function for the population is characterized by too small standard deviation ($\delta(F(t))$) in relation to the span of the fitness function values ($max(F_{av}(t)-F_{min}(t), F_{max}(t)-F_{av}(t))$), then the desired operation is to increase in the probability of histogram selection (third position in the formula (14)). Otherwise, it desired to increase the probability of selection by a deterministic roulette method (first position in formula (14)). If the parameters of the population are within the range considered preferable, the probabilities of selection of both methods are nearly equal (the second entry in the formula (14)). It should be noticed that the rule must always be fulfilled: $p_{his}(t)+p_{det}(t)=1$, i.e., any of the methods must always occur.

4.3. Interval selection

This method is based on the division of the parental population into several subpopulations. The criterion for the division are values of the fitness function of individuals. At the beginning, a sorted list of individuals with different values of fitness function in a population is created. In many cases, equal values of the fitness function may characterize different solutions (individuals). This selection method deals with such a case and additional items are created on the list, including different individuals with the same values of fitness function. A whole range of values of this function occurring is divided into some number of compartments. This number of compartments is one of the parameters of the method. The most common distribution used in simulations contains three subpopulations, with the following parameters:

- individuals assessed at 90–100% of the best value of the fitness function;
- individuals assessed at 50–90% of the best value of the fitness function;
- individuals assessed at 10–50% of the best value of the fitness function.

Each compartment is guaranteed a certain percentage of individuals that will appear in the next parent population. Of course, the worse the interval of fitness function values, the fewer should be a guaranteed percentage of the next parent population participation.

Percentage distribution of the number of descendants for each compartment can also be adapted to the requirements of the problem being solved. In the conducted computer simulations, the best results were obtained using the following schedule:

- The first interval (the best individuals) receives 60% of the descendant population.
- The second interval (average individuals) provides 30% of individuals of the descendant population.
- The third interval (weakest individuals) receives 10% of the descendant population.

The sum of all pools of the intervals forms the entire new population. The selection of individuals that pass to the next population within a compartment may be carried out in various ways. In the presented results of the computer simulations, the selection of the best individuals was used, but of course, several well-known methods for instance tournament selection or other conventional methods, can be used.

The applied distribution of the parent population (the number of used compartments and their percentage distribution) and allocated pools of seats to be filled in the next population are method parameters and can be modified, not only a priori but also during the algorithm run-time. With such modifications, it is possible to change the profile of the desired descendant population and significantly modify the properties of the method of selection of individuals.

It should be noted that this method is applicable in cases where the descendant portion of the population (λ) is greater than the parent part (μ) or selection is made from both parts ($\mu + \lambda$) to have a suitable subject pool from which to select.

4.4. Selection with lifetime and taboo list

The problem of stagnation in the evolutionary computation after reaching a certain level of solutions associated with the stagnation in local optima is widely known and very difficult to control in evolutionary algorithms. There are many ways to deal with this problem, from "killing" all parent individuals (even if are better than their offspring) in non-elitist methods, through methods that allow continuous monitoring of the diversity of the population and insert the newly drawn solutions to maintain a high diversity of the population to controlled selection methods. The presented in this point method is a kind of controlled selection, combining several previously used solutions in different areas of artificial intelligence in one – a selection using the lifetime of an individual and an array/list of taboo solutions. The lifetime of an individual was already proposed in slightly different versions – as an equivalent of the selection method in the EA with varying population sizes in the work Arabas [2]. The taboo list is used in several versions of methods for solving optimization problems called taboo search, where it is used to store a variety of information about the discarded for some time solutions, their components, or methods of modification of solutions. This new method of selection can best be characterized as follows: the method of selection of individuals to the next parent population can be any previously known method

of selection, while the proposed method is responsible for the initial preprocessing of the population.

The creation of a new solution as a result of the random generation of the initial population, or as an effect of reproduction using genetic operators, is associated with giving it a certain value, which is a maximum lifetime in iterations (generations) of the evolutionary algorithm. This value may depend on the quality of an individual, but it is not required. It is not a guaranteed lifetime, the solution may eliminate earlier the selection method used. During its lifetime, the solution can contribute to the creation of new solutions, but maximally after its lifetime, it is eliminated from the population and can be inserted into the taboo list. It enrolls solutions with appropriate parameters: they must have good values of the fitness function (in the range of 60–100% of the current best) and cannot be too similar to those already on the list (in terms of content code of compared solutions). The issue of assessing the similarity of solutions has already been mentioned while describing the histogram selection, here it looks the same, but the similarity of solutions placed on the taboo list has to be even more distant (in the Hamming sense, for instance). It contains rather certain classes or patterns of similar solutions, i.e. not solutions with differences in the one position, but bigger. It must be noticed that with the taboo list also rejected solutions have a certain effect on the form of a population of solutions because any solution that is too similar (in this case the differences of similarity threshold may be smaller than when placed on the list) to the existing on the taboo list is eliminated from the population, even if it is still "young". The taboo list is reset after finding the next best solution. The taboo list becomes active during the stagnation of calculations. If evolutionary computations frequently lead to the discovery of new and better solutions, the list remains inactive. This allows to "beat in" the population from a local optimum if necessary, requiring no additional effect and delay of the calculation, when it is not needed.

5. Properties of proposed methods in computer simulations

Theoretical formulas for coefficients measuring selection characteristics are rather difficult to obtain for more complicated methods, although, there are some results for standard ones [6, 10, 15, 16, 22]. Thus, a set of practical simulations can show results obtained for the selection methods presented in this study, based on solved computational problems. This method of properties investigation is not such universal as derived formulas, but despite all, can tell a lot about the properties of proposed selection methods.

5.1. Sample computational problems for testing described selection methods

As a testing base, an evolutionary algorithm solving the maximum α-clique searching problem has been chosen. An α-clique is a generalization of a clique notion. The clique

in a graph is every complete subgraph of the whole graph. The complete subgraph is a subgraph where all vertices have edges between them. For sparse graphs, cliques are rather small and are too highly connected structures to find locally proper connected centers, for instance, real transportation networks. Some kind of more flexible and controllable structure would be better to divide the whole graph and find locally interconnected structures. This elementary brick for graph partitioning could be an α-clique. The α-clique is defined in [18] or [14]:

Let $A = (V', E')$ be a subgraph of graph $G = (V, E)$, $V' \subseteq V$, $E' \subseteq E$, $k = Card(V')$ and let k_i be a number of vertices $v_j \in V'$ that v_i, $v_j \in E'$.

1. For $k = 1$ the subgraph A of graph G is an α-clique with desired value of α.
2. For $k > 1$ the subgraph A of graph G is an α-clique with the desired value of α if for all vertices $v \in V'$ fulfill the condition $\alpha \leqslant \frac{k_i+1}{k}$, where $\alpha \in (0, 1]$.

The α-clique is simply a subgraph, where all nodes are connected with not less than $\alpha \cdot 100\%$ of all their nodes (the subgraph size), with α representing the desired connectivity percentage. Of course, the α-clique with $\alpha = 1$ is simply a clique.

From fundamental basic graph properties, it can be deduced that for $\alpha > 0.5$ obtained α-cliques must constitute connected subgraphs. A connected (sub)graph is a kind of graph, where for each pair of vertices there is a path (a continuous sequence of edges) between them and this is the most interesting case because it is not good to derive transportation centers (often called hubs) with isolated, not connected vertices.

Similarly to the clique case, also it can be useful to find the maximum α-clique in a graph. It is a non-trivial task, practically harder than finding the maximum clique problem (the problem of finding the maximum clique is NPH [1]), because not every subgraph of α-clique with imposed α is an α-clique with the same value of α, very often the value of α in a sub-α-clique is lower than in the whole α-clique. In the case of the clique, all subgraphs of this clique are also cliques. Thus, it is difficult to prepare a simple algorithm trying to find the biggest α-clique by adding nodes to the obtained one, because one can never know how many and which of them must be added. In the case of a clique, one doesn't know only which one may be added, if it is possible to make it bigger, but if it is possible, bigger cliques can be obtained by adding one new vertex in one step. In the case of α-clique, this is often not possible. This fact means that the majority of efficient approximate algorithms prepared to find the maximum cliques cannot be extended to search for bigger or maximum α-cliques. Thus, it seems justified to use the evolutionary algorithm to solve the problem, because this method can do it efficiently.

As the second computational test problem, the widely known classical TSP was considered. It was used only in several cases to show some interesting properties of selection methods that led to the observation that properties of selection methods also depend on the solved problem.

5.2. Evolutionary algorithm used to solve the problem

5.2.1. The maximum α-clique problem

The information about the considered graph is stored in a square neighborhood matrix that describes connections of all graph nodes: 0 – no connection, 1 – presence of connection. Because a single node is also treated as an α-clique, the matrix has 1 on the main diagonal. The value of α (the α-clique parameter) is imposed as the problem parameter.

Each member of the EA population encodes the solution to the problem as a variable-length vector representing the biggest α-clique derived by that member. The not selected nodes form a similar vector, serving as a repository of potential new nodes that can be attached to the derived α-clique. Additionally, each member of the population contains more data. This includes a vector of real numbers, which describes its knowledge about genetic operators and the operator number chosen to modify the solution in the current iteration. More details about genetic operators and the method of evaluation, selection, and application of them will be given later in this chapter.

The described data structure requires specialized genetic operators, which modify the population of solutions. Each operator is designed in such a manner that it preserves the property of being an α-clique with the desired value of α for the modified solutions (after its application, the actual value of α for the modified solution is computed). If a modified solution violates the limitation of being an α-clique, the operation is canceled and no modification of the solution is performed. While this method makes it more challenging for the evolutionary algorithm to find satisfactory solutions, potentially encountering greater difficulties with local extrema compared to, for instance, a method with a penalty function, it ensures that the computed solutions are admissible.

Designed genetic operators are:

- mutation – an exchange of randomly chosen nodes in α-clique and the storage vector,
- transfer of randomly chosen node from the storage to the α-clique,
- "intelligent" movement – performed only if this modification gives a better value of fitness function,
- concatenation – this operator tries to concatenate small vectors chosen from the storage with α-clique,
- multiple versions of operators are also applied (randomly selected numbers of repetitions of the genetic operator in one generation).

5.2.2. The TSP problem

Solutions in this case are encoded as lists of cities to be visited in the order described by the list. Several genetic operators were used to solve the problem: random and

heuristic ones. All operators ensure that the generated offspring can consistently encode feasible solutions. The operators are:

- mutation – a random exchange of two cities in the list of cities,
- crossover – starting from the first city of one list, the next cities are chosen in turn from one or second list (the city closer to the previously accepted city is selected from the parent individual and introduced to the offspring),
- inversion – a randomly chosen fragment of the list is used in the reverse order,
- transposition – a randomly chosen fragment of the list of cities is moved to another place (also randomly chosen) in the list,
- 2-optimal method – exchange of two chosen edges in the route, if it gives a shorter route (based on the widely used k-optimal method [21] for approximate solving of TSP),
- neighborhood-1 operator – exchanges a randomly chosen city in the route for another, randomly chosen from the list of the closest ones in the geometric sense (a list of several closest cities is generated for every city during the initialization of the algorithm),
- neighborhood-2 operator – takes a city close to the selected one from the path, moves all cities between them by one position, and inserts the picked city next to the selected one.

5.2.3. The common part of used evolutionary algorithms

The application of several specialized genetic operators requires a method that selects and executes them during evolutionary computations. In the approach used in [17], it is assumed that only one, selected operator modifies the solution in one generation (not two as in typical EA). In that case, it is easy to allocate the result of that operation to the individual and operator. Thus, the operator obtaining better results should have a higher probability and more frequently affect the population than the worse one. But it is very likely that the operator, that is proper for one individual, gives worse effects for different solutions because of its location in the search space.

Thus, each individual may have its preferences, enabling it to select operators that align with its specific characteristics or requirements. For example, an individual might favor operators that excel in certain regions of the search space or demonstrate effectiveness for specific types of solutions. To obtain this possibility, every individual has a vector of quality factors, where each factor corresponds to one genetic operation and is a measure of the quality of that operator. The normalized vector of qualities becomes a base to compute the probabilities of selection and execution of genetic operators by population members (15).

This set of probabilities – a base of population experience can be inherited and improved over the next generations.

$$p_{ij}(t) = \frac{q_{ij}(t)}{\sum_{i=1}^{L(t)} q_{ij}(t)} \tag{15}$$

where:

p_{ij} – the probability of execution of the genetic operator,
$q_{ij}(t)$ – a quality factor of the genetic operator,
$L(t)$ – the actual number of genetic operators (in some evolutionary algorithms may vary during computations),
t – the current time.

The method to compute the quality factors is based on reinforcement learning [8, 20] (one of the algorithms used in machine learning). An individual is treated as an agent whose role is to select and call one of the evolutionary operators. When the selected i-th operator is applied, it can be regarded as an agent's action a_i leading to a new state s_i which in this case is a new, modified solution. The agent receives a reward or a penalty respective to the quality of the new state (solution). The aim of the agent is to perform the actions that give the highest long-term discounted cumulative reward V^*. The described activity leads to the Q-learning technique of temporal reward assignments, which can be written as:

$$V(s_{t+1}) = V(s_t) + \beta(r_t + \gamma V^*(s_{t+1}) - V(s_t)) \tag{16}$$

where:

$V(s_t)$ – a quality factor or discounted cumulative reward, that can be identified with q_{ij} from (15),
$V^*(s_{t+1})$ – estimated value of the best quality factor (in the experiments the value gained by the best operator was taken),
β – a learning factor,
γ – a discount factor,
r_t – the reward for the best action, which is equal to the improvement of the quality of a solution after execution of the evolutionary operator,
t – index of the current moment in time.

Selection methods used in the presented evolutionary algorithm to obtain shown further results were described in sections 3 and 4.

5.3. The testing problem data

Unfortunately, it is not possible to obtain the testing data (data with known optimal solution) for the maximum α-clique problem from any data repositories. Instead, it is possible to find several problems for the maximum clique. Since the clique is a special case of α-clique with $\alpha = 1$, the problems for the maximum clique found in BHOSLIB: Benchmarks with Hidden Optimum Solutions for Graph Problems (Maximum Clique, Maximum Independent Set, Minimum Vertex Cover and Vertex Coloring) – Hiding Exact Solutions in Random Graphs [4] have been used. The chosen problem was

a graph with 4000 vertices and 7,425,226 edges with the maximum clique equal to 100 (frb100-40.clq.gz). In all cases, starting populations are generated using a simple greedy algorithm, which randomly chooses the first node and tries to add to the obtained α-clique remaining graph nodes in random sequence. This method is, as it was said, rather a poor optimization method but is quite fast and generates different solutions with a size of about 70% of the size of known the maximum clique, which can be a good starting point for the evolutionary search of better ones.

The second considered problem is a classic TSP problem with 1002 cities (pr1002.tsp) [9] with the optimal solution 259,045. In this case, also a greedy algorithm was used to generate the initial population of solutions.

5.4. Results obtained for selected coefficients of selection characteristics

5.4.1. The takeover time

The takeover time shows the strength of the selective pressure of a selection method, showing how fast the whole population would converge to one, the best solution, assuming the lack of genetic operators and preserving one copy of the best individual. For simpler methods (proportional/roulette selection, deterministic roulette, tournament selection) this time can be described analytically, but for more complicated ones this is difficult or maybe not possible. Thus, in this work, values obtained from real computer simulations are presented. Table 1 presents averaged results obtained from 10 simulations. Simulations lasted maximally 100 epochs, if during this time the population hadn't converged, the takeover time was assumed to an infinite.

Table 1
Obtained takeover times for investigated selection methods

Selection method	Roulette / Roulette with elitism*	Deterministic roulette	Tournament	Histogram	Mixed: deterministic + histogram
Takeover time (epochs)	15.6	7.8	9.4	∞	6.8
Selection method	Histogram flat	Mixed: deterministic + histogram flat	Interval	Mixed: deterministic + histogram flat with lifetime and taboo	–
Takeover time (epochs)	∞	6.2	∞	8.8	–

*In this case it is the same selection method, the investigation of takeover time assumes preserving the best individual in the population (elitism).

As it can be seen, selections with the heuristic population control mechanism (histogram, histogram flat, and interval) have infinite takeover time, traditional ones have takeover times from 6.8 (roulette) to 9.4 (tournament), mixed selections have takeover time similar to the component with stronger selection pressure, but this value depends on the probability of execution of both component methods, which can be seen in Table 2.

Table 2
Obtained takeover times for different probabilities of component selection methods in mixed selection

	Mixed: deterministic + histogram	Mixed: deterministic + histogram flat	Mixed: deterministic + histogram flat with lifetime and taboo
$p_{his} = 0.95$	∞	34.4	∞
$p_{his} = 0.85$	41.0	10.8	29.6
$p_{his} = 0.75$	29.4	11.6	15.8
$p_{his} = 0.65$	19.0	7.8	11.4
$p_{his} = 0.55$	14.4	7.8	9.0
$p_{his} = 0.45$	6.8	6.2	8.8

For the selection with lifetime and taboo list, the value of this parameter is not a proper measure of selection properties, because for established longer lifetime than the takeover time (this situation is presented in Tables 1 and 2, where the lifetime is set to 100), there is no influence of it on the obtained takeover value. For shorter than takeover lifetimes imposed, all the population members should be killed (no new individuals are created since the genetic operators are disabled) after this time and only the best individual is artificially preserved to fulfill the conditions of the coefficient measure, thus the takeover time is equal to the imposed maximal lifetime of individuals. This is not the typical situation that this selection method works and obtained in this case value would say nothing about this selection properties. This selection method is prepared for long-time computations and in this case, its properties can be observed. Thus, the simple conclusion is that the takeover time is not a good measure of properties for more sophisticated selection methods.

5.4.2. The loss of diversity caused by the selection process

The *loss of diversity* is a harmful phenomenon, especially in small populations (such as typically used in evolutionary computations), because it can lead to the premature convergence of the population to local minima. If the mutation level is rather small, a crossover of almost identical individuals produces almost the same individuals and the progress in computations is negligible. Even if more sophisticated operators are used, similar phenomena occur. But of course, selection is necessary to continue the

process of evolution and better individuals should have more descendants than worse. As in other aspects of evolutionary algorithms, it is important to keep the balance between the diversity of the population removing the worst and replication the best solutions. Unfortunately, the formula for the optimal composition of the population is not known. We can only make experiments with different types of selection (see Fig. 1) and evaluate them, comparing obtained results for the solved problem. But this would allow only to see which one is better for the given instance of the solved problem.

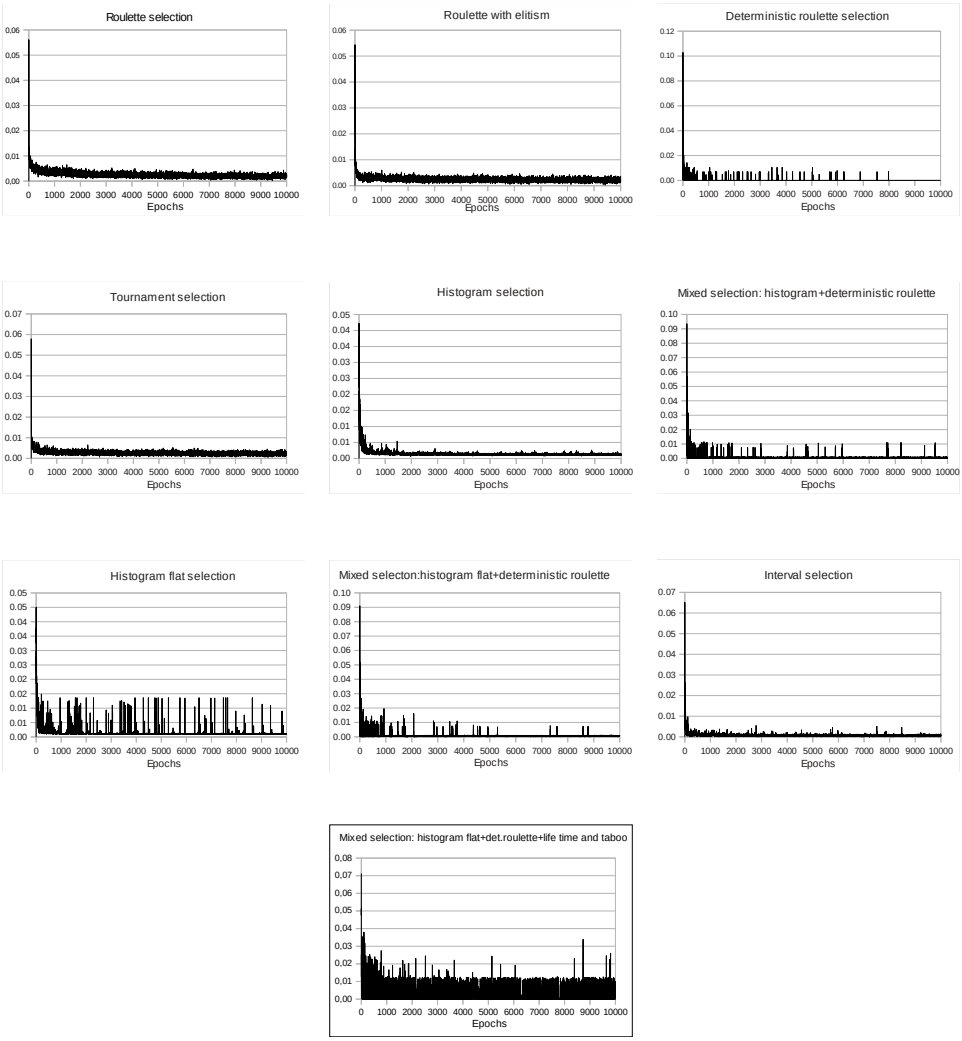


Figure 1. Graphs of average values of the loss of diversity factor for several tested selection methods

Table 3
The comparison of the average value of the loss of diversity factor in the maximum α-clique problem using tested selection methods

Selection method				
Roulette	Roulette with elitism	Deterministic roulette	Tournament	Histogram
0.00252	0.00244	0.00017	0.00259	0.00148
Mixed: deterministic + histogram	Histogram flat	Mixed: deterministic + histogram flat	Interval	Mixed: deterministic + histogram flat with lifetime and taboo
0.00075	0.00139	0.00070	0.00076	0.00128

The values collected and presented in Table 3 are the average values of 10 simulations (10,000 epochs) recorded at each epoch using formula (4). The common part of populations considers not only the fitness function value but also the encoded solutions. Individuals with identical values of fitness function but encoding different solutions are treated as different.

As it can be noticed, the smallest values of the *loss of diversity* factor (the most similar populations before and after selection) have: a mixed selection consisting of histogram flat and deterministic roulette with lifetime and taboo, interval selection, and roulette selection, but only the first one gives also very good results in problem-solving (see Tables 4 and 5). It suggests that small values of the *loss of diversity* factor are important, but it is not the most decisive factor that influences the evolution process.

5.4.3. The selection intensity

The selection intensity factor shows the convergence properties of the investigated methods. For simpler ones, there are some theoretical results [7, 13], for more sophisticated and heuristic selection methods only practical experiments can show their behavior. The results obtained during simulations are collected in Figure 2.

As can be noticed, several selection methods like roulette with elitism, deterministic roulette, tournament selection, histogram selection, histogram flat selection and mixed selections consisting of histogram ones with deterministic roulette have the property of converging their selection intensity at a value close to 0.2, but roulette selection, interval selection, and mixed selection consisting of histogram flat, deterministic roulette with lifetime and taboo do not converge. It means that there are strong changes in population composition during the whole algorithm simulation, while the converging ones have rather small changes.

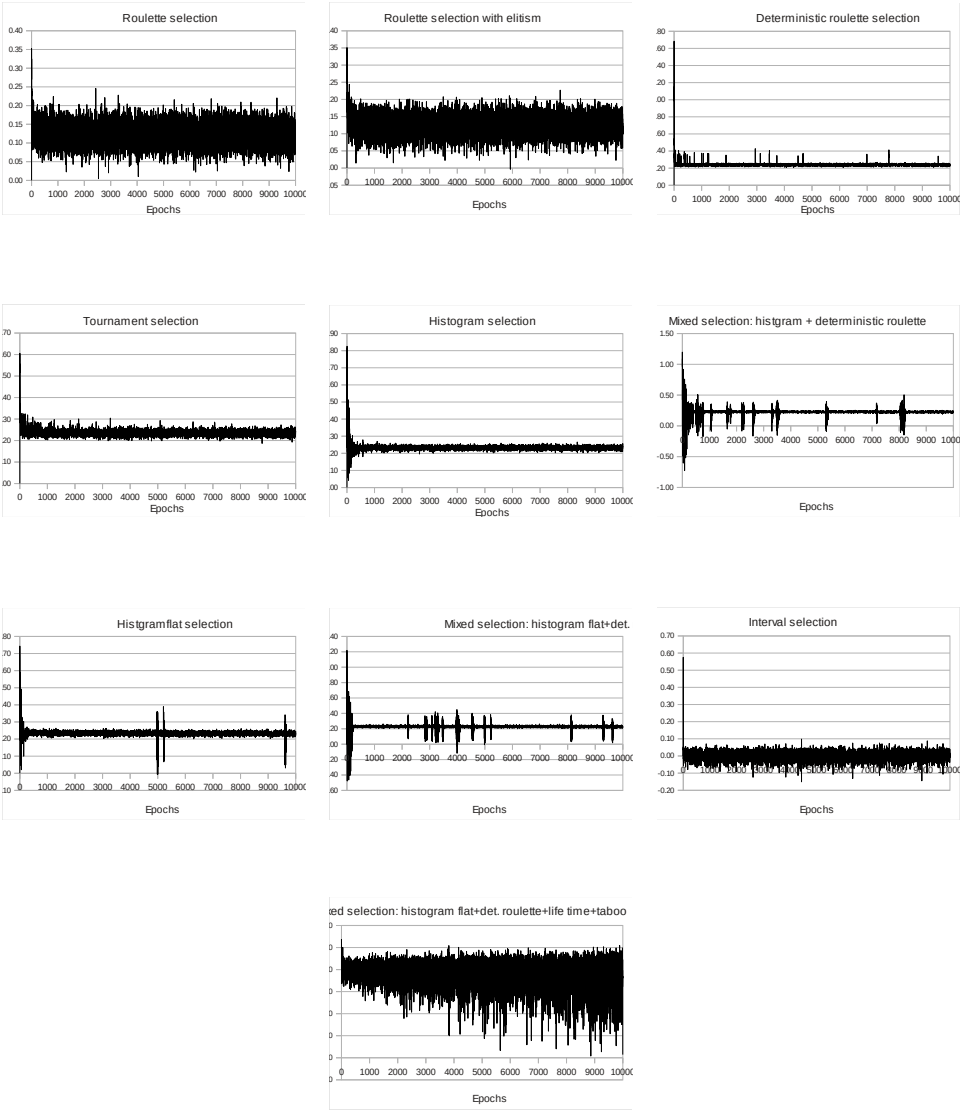


Figure 2. Graphs of average values of the selection intensity factor for several tested selection methods

5.4.4. The selection variance

The *selection variance* (3) also describes the properties of selection methods, in this case, it is possible to trace how changes the variance of evaluations of solutions in the population as a result of the selection process. The obtained results are presented in Figure 3.

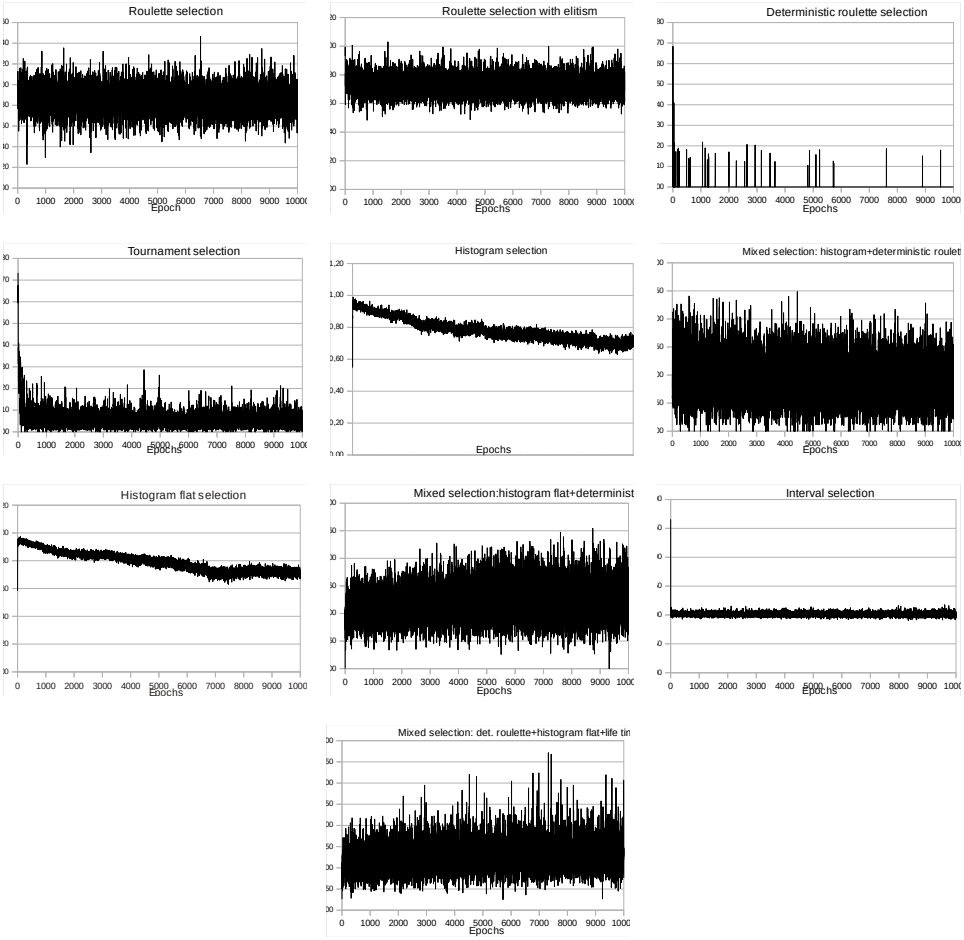


Figure 3. Graphs of average values of the selection variance for several tested selection methods

5.4.5. The population diversity before (s_b) and after (s_a) selection

The diversity of the population before and after selection can also be a measure of its properties. It is widely known that higher population diversity favors better results achieved by the evolutionary algorithm due to the prevention of premature convergence of the population of solutions. The presented further results were obtained for the maximum α-clique problem with $(\mu+\lambda)$ strategy[3] of the population development with $\mu = 100$ and $\lambda = 700$. The average values of population diversity as defined in (5) and (6) are presented in Table 4.

[3]It means that the new parent population was selected from the old parent and offspring populations.

Table 4

Values of average population diversity coefficients s_b and s_a obtained using investigated selection methods for the maximum α-clique problem

	Selection method				
	Roulette	Roulette with elitism	Deterministic roulette	Tournament	Histogram
s_b	0.0222	0.0200	0.1052	0.0215	0.1332
s_a	0.0946	0.0844	0.7834	0.0912	0.9998
	Mixed: deterministic + histogram	Histogram flat	Mixed: deterministic + histogram flat	Interval	Mixed: deterministic + histogram flat with lifetime and taboo
s_b	0.1302	0.1331	0.1286	0.1364	0.0661
s_a	0.9838	1.0000	0.9690	0.8948	0.4865

Table 5

Results of EA simulations obtained using investigated selection methods for the maximum α-clique problem (average sizes of obtained α-cliques)

Iteration	Selection method				
	Roulette	Roulette with elitism	Determinstc roulette	Tournament	Histogram
0	70.7	70.5	70.9	71.0	70.7
10	71.1	71.3	72.0	71.4	73.2
100	73.0	74.4	77.1	74.2	77.6
1000	78.4	78.5	81.8	79.6	81.1
10,000	84.0	81.8	85.9	84.3	84.2
100,000	87.5	82.7	87.6	88.0	85.6
Iteration	Mixed: deterministic + histogram	Histogram flat	Mixed: deterministic + histogram flat	Interval	Mixed: deterministic + histogram flat with lifetime and taboo
0	70.8	70.7	70.4	70.5	70.8
10	73.4	72.2	72.4	71.7	72.1
100	78.8	75.7	79.9	75.5	77.4
1000	84.7	79.0	85.3	78.4	84.3
10,000	87.7	81.3	88.4	80.5	88.4
100,000	88.3	82.8	88.9	83.6	91.0

Comparing these values with obtained solutions (Tab. 5) it can be seen that the highest values of population diversity (histogram flat, histogram) are not necessarily bounded with the best results obtained. The same is true for the lowest values of population diversity (see Fig. 4).

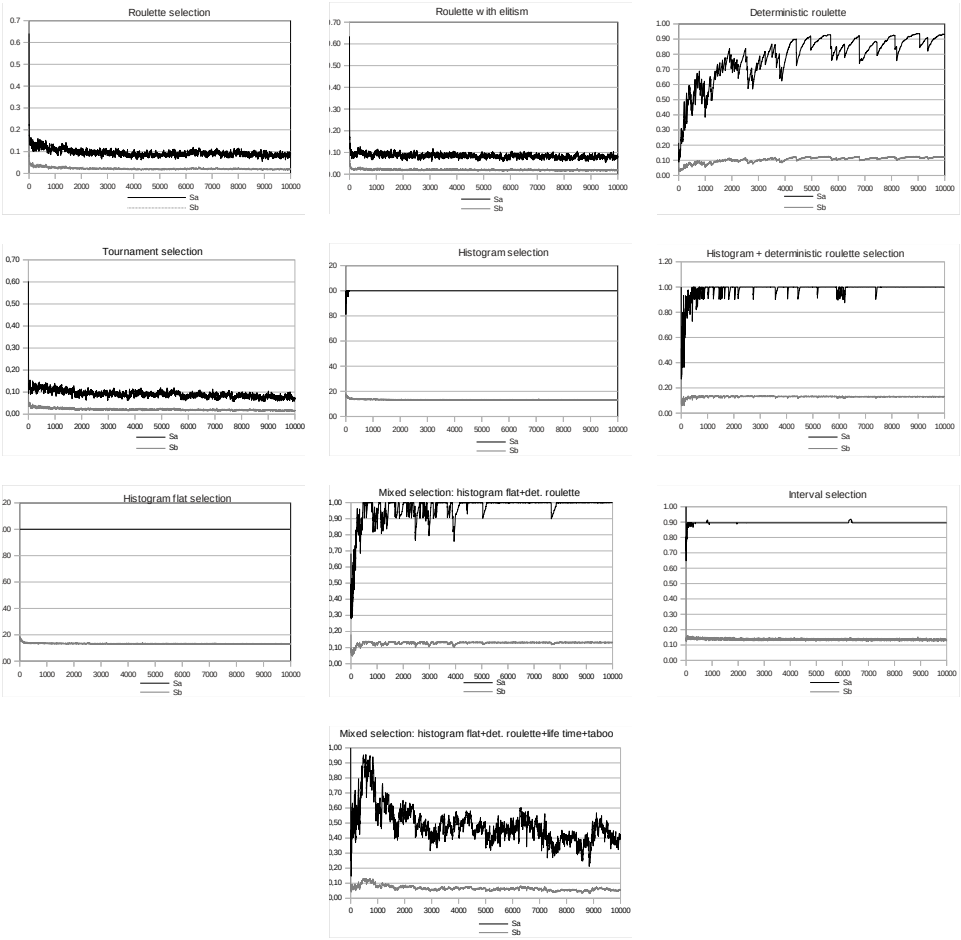


Figure 4. Graphs of average values of the diversity before (s_b) and after (s_a) selection for several tested selection methods

5.5. Values of obtained solutions for solved problems using investigated selection methods

Tables 5 and 6 show the averaged results of 10 computer simulations for several selected computation stages. As it can be seen, mixed selections outperform all other methods and the best of them is mixed selection with lifetime and taboo (Tab. 5). Differences among them are not very big, but noticeable. Similar conclusions can be drawn by analyzing Table 6, where the best results are obtained using mixed selection consisting of deterministic roulette and histogram flat selections, but the version with lifetime and taboo gives only slightly worse results.

The worst results are obtained using the roulette method (no improvement of obtained results has been recorded) and interval selection (Tab. 5), for the TSP problem (Tab. 6) the roulette method is significantly worse than other methods. This is not a surprise, because the roulette method is rather used for theoretical, not practical purposes. The interval selection method has been prepared for non-stationary optimization tasks and in this kind of optimization problem it works very well (see [19]) but it is not as good for stationary problems.

Table 6
Results of EA simulations obtained using investigated selection methods for the TSP problem (average distances)

Iteration	Selection method				
	Roulette	Roulette with elitism	Determinstc roulette	Tournament	Histogram
0	292,616.4	293,119.9	292,608.4	292,511.5	292,038.7
10	292,616.4	292,926.4	291,143.0	292,511.5	291,178.2
100	292,616.4	290,326.4	281,564.4	292,511.5	281,264.4
1000	292,616.4	281,528.9	272,017.7	292,511.5	271,265.7
10,000	292,616.4	274,079.4	271,535.8	275,166.7	270,780.7
100,000	292,616.4	272,065.9	270,918.2	268,827.5	270,320.8
Iteration	Mixed: deterministic + histogram	Histogram flat	Mixed: deterministic + histogram flat	Interval	Mixed: deterministic + histogram flat with lifetime and taboo
0	291,958.8	292,427.0	292,124.7	292,529.8	292,003.8
10	290,503.3	291,543.8	291,140.0	292,125.2	290,449.4
100	279,740.3	283,248.2	282,115.1	285,322.9	281,310.9
1000	270,646.6	271,432.7	271,620.0	273,924.4	271,636.1
10,000	270,085.0	270,096.3	270,033.5	270,089.8	270,287.9
100,000	269,598.5	269,728.8	268,197.1	268,420.4	268,291.6

The comparison of average computation times of computer simulations (Tab. 7) shows that new selection methods are rather fast and also the times of their operation are competitive with those used traditionally.

Table 7
The comparison of average computation times (in seconds) of evolutionary simulations duration using tested selection methods for solved problems

Solved problem	Selection method				
	Roulette	Roulette with elitism	Deterministic roulette	Tournament	Histogram
α-clique	234,796	184,709	219,306	214,365	211,530
TSP	1160.4	1247.3	761.4	1114.5	1176.6
Solved problem	Mixed: deterministic + histogram	Histogram flat	Mixed: deterministic + histogram flat	Interval	Mixed: deterministic + histogram flat with lifetime and taboo
α-clique	173,653	189,012	186,148	188,338	83,099
TSP	1048.7	1012.7	775.2	7080.7	1103.0

6. Conclusions

The selection in nature is the main (and maybe even the only) force that directs the population development. Effects of its slow and simple but powerful activity can be seen everywhere, admiring the variety of animal and plant species. Artificial selection methods, which are usually used in evolutionary algorithms, have far less time to achieve the desired results, although the results don't have to be so spectacular as in the case of the natural one, but they should be quick and effective. Unfortunately, selection methods, usually used in EA, are mainly quite simple because they try to mimic the natural one, without any possibilities of tuning, control, learning, and adaptation. Presented in this article methods try to overcome this problem and improve this important part of almost every EA to enable more predictable, resistant to local optima, flexible, and faster behavior of used selection methods.

References

[1] Aho A.V., Hopcroft J.E., Ullman J.D.: *The Design and Analysis of Computer Algorithm*, Addison-Wesley Publishing Company, 1974.

[2] Arabas J., Michalewicz Z., Mulawka J.: GAVaPS – a genetic algorithm with varying population size. In: *Proceedings of The First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, vol. 1, pp. 73–78, IEEE, 1994. doi: 10.1109/ICEC.1994.350039.

[3] Back T.: Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, vol. 1, pp. 57–62, IEEE, 1994. doi: 10.1109/ICEC.1994.350042.

[4] BHOSLIB – Network Data Repository. http://www.nlsde.buaa.edu.cn/kexu/benchmarks/graph-benchmarks.htm.

[5] Blickle T., Thiele L.: *A Comparison of Selection Schemes used in Genetic Algorithms*, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), 1995. TIK-Report Nr. 11.

[6] Blickle T., Thiele L.: A Mathematical Analysis of Tournament Selection. In: *Proceedings of the 6th International Conference on Genetic Algorithms*, vol. 1, pp. 9–16, Morgan Kaufmann Publishers, San Francisco, 1995.

[7] Cantú-Paz E.: Selection Intensity in Genetic Algorithms with Generation Gaps. In: *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation (GECCO'00)*, vol. 1, pp. 911–918, Morgan Kaufmann Publishers, San Francisco, 2000.

[8] Cichosz P.: *Systemy uczące się*, WNT, Warszawa, 2000.

[9] ELIB: MP-TESTDATA – The TSPLIB Symmetric Traveling Salesman Problem Instances. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html.

[10] Goldberg D.E., Deb K.: A comparative analysis of selection schemes used in genetic algorithms, *Foundations of Genetic Algorithms*, vol. 1, pp. 69–93, 1991. doi: 10.1016/B978-0-08-050684-5.50008-2.

[11] Holland J.J.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, USA, 1992.

[12] Moscato P.: *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*. Technical Report C3P 826, Caltech Con-Current Computation Program 158-79, California Institute of Technology, Pasadena, 1989.

[13] Muhlenbein H., Schlierkamp-Voosen D.: Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization, *Evolutionary Computation*, vol. 1(1), pp. 25–49, 1993. doi: 10.1162/evco.1993.1.1.25.

[14] Potrzebowski H., Stańczak J., Sęp K.: Separable Decomposition of Graph Using α-*cliques*. In: M. Kurzynski, E. Puchala, M. Wozniak, A. Zolnierek (eds.), *Computer Recognition Systems 2*, Advances in Soft Computing, vol. 45, pp. 386–393, Springer, Berlin–Heidelberg, 2007. doi: 10.1007/978-3-540-75175-5_49.

[15] Rogers A., Prugel-Bennett A.: Genetic Drift in Genetic Algorithm Selection Schemes. In: *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 298–303, 1999. doi: 10.1109/4235.797972.

[16] Rudolph G.: Takeover Times and Probabilities of Non-Generational Selection Rules. In: *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation*, pp. 903–910, Morgan Kaufmann Publishers, San Francisco, 2000.

[17] Stańczak J.: Biologically inspired methods for control of evolutionary algorithms, *Control and Cybernetics*, vol. 32(2), pp. 411–433, 2003.

[18] Stańczak J., Potrzebowski H., Sęp K.: Evolutionary approach to obtain graph covering by densely connected subgraphs, *Control and Cybernetics*, vol. 40(3), pp. 849–875, 2011.

[19] Stańczak J., Trojanowski K.: Non-stationary optimization with multi-population evolutionary algorithm. In: J. Arabas (ed.), *Evolutionary Computation and Global Optimization*, pp. 251–260, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2007.

[20] Sutton R.S., Barto A.G.: *Reinforcement Learning: An Introduction,* 1st ed., MIT Press, USA, 1998.

[21] Sysło M.M., Deo N., Kowalik J.: *Discrete Optimization Algorithms with Pascal Programs*, Dover Publications, USA, 2006.

[22] Thierens D., Goldberg D.: Convergence Models of Genetic Algorithm Selection Schemes. In: Y. Davidor, H. Schwefel, R. Männer (eds.), *Parallel Problem Solving from Nature – PPSN III,* Lecture Notes in Computer Science, vol. 866, pp. 119–129, Springer, Berlin–Heidelberg, 1994. doi: 10.1007/3-540-58484-6_256.

[23] Zhong J., Hu X., Zhang J., Gu M.: Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms. In: *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 2, pp. 1115–1121, IEEE, 2005. doi: 10.1109/CIMCA.2005.1631619.

Affiliations

Jarosław Stańczak
Systems Research Institute PAS, Newelska 6, 01-447 Warsaw, stanczak@ibspan.waw.pl

Received: 31.03.2023
Revised: 18.12.2023
Accepted: 18.12.2023

Dariusz Mikolajewski, Anna Bryniarska
Piotr Michal Wilczek, Maria Myslicka
Adam Sudol, Dominik Tenczynski
Michal Kostro, Dominika Rekawek
Rafal Tichy, Rafal Gasz
Mariusz Pelc, Jaroslaw Zygarlicki
Michal Koziol, Radek Martinek
Radana Kahankova Vilimkova, Dominik Vilimek
Aleksandra Kawala-Sterniuk

THE MOST CURRENT SOLUTIONS USING VIRTUAL-REALITY-BASED METHODS IN CARDIAC SURGERY – A SURVEY

Abstract *There is a widespread belief that VR technologies can provide controlled, multisensory, interactive 3D stimulus environments that engage patients in interventions and measure, record and motivate required human performance. In order to investigate state-of-the-art and associated occupations we provided a careful review of 6 leading medical and technical bibliometric databases. Despite the apparent popularity of the topic of VR use in cardiac surgery, only 47 articles published between 2002 and 2022 met the inclusion criteria. Based on them, VR-based solutions in cardiac surgery are useful both, for medical specialists and for the patients themselves. The new lifestyle required from cardiac surgery patients is easier to implement thanks to VR-based educational and motivational tools. However, it is necessary to develop the above-mentioned tools and compare their effectiveness with Augmented Reality (AR). For the aforementioned reasons, interdisciplinary collaboration between scientists, clinicians and engineers is necessary.*

Keywords virtual reality, cardiac surgery, clinical applications, surgical training, image processing

Citation Computer Science 25(1) 2024: 123–145

1. Introduction

The enthusiasm of the use of virtual reality (VR) in cardiac surgery is relatively short, although, the first scientific publications on the subject appeared in 2002 [17, 40]. However, the growing interest and importance of the combination of the above mentioned technologies for the future of interventional cardiac surgery and the imaging of cardiovascular function in the form of virtual twins puts the subject at the centre of interest for engineers and clinicians alike [81]. In addition to preventive medicine or bespoke cardiac interventions, this includes regenerative surgery in the form of the emerging possibility of 3D printing tissues from bio-ink (in most cases: the patient's stem cells) through reverse engineering (3D scan – modification – 3D printing) [26, 92]. We are already able to print other human tissues, such as skin (innervated and vascularised), but larger and more complex organs such as the pancreas, liver, lungs or just the heart have so far been beyond our reach. The development of VR in cardiac surgery could be a good step in this direction, heralding new possibilities, including those based on the Internet of Things, integrating them into the Healthcare 4.0 paradigm and expanding the capabilities of the cardiac surgeon [45, 98].

The main aim of this paper is to summarise the current and emerging future opportunities for VR-based support of cardiac surgery, also with reference to the observed Virtual-Augmented Reality (VR-AR) rivalry in both, industrial (Industry 4.0) and clinical applications [70, 87].

The most recent technological advances inevitably bring medical improvements, giving professionals increasingly effective diagnostic, therapeutic, rehabilitation and care tools, including automated and semi-automated long-term care [25, 54, 78]. Placing this article in a broader context, we can see immediately that this is highly interdisciplinary research, combining technical sciences (computer science, mechatronics, material engineering) with medical and health sciences (including not only cardiology, but also medical imaging, biotechnology, and tissue engineering), as well as the humanities and social sciences (including psychology) [21, 33]. The relevance of the above-mentioned research goes beyond its scientific and clinical context, being of great economic and social importance, both towards increasing health-related quality of life (HRQoL) and making it easier for people with cardiovascular conditions to learn, work and play, as well as ageing more cheerfully [56, 77, 100].

In this paper, we propose not only an overview of the current state of the art, but will carefully consider a number of hypotheses for the further development of this area of knowledge and practice, including a better understanding of the physiological and pathological mechanisms of the cardiovascular system, advances in didactics on virtual cardiac simulations, the role of medical simulation in the training of cardiac surgeons, the preparation of cardiac procedures (e.g. analysis of access routes) in the case of a specific patient, not shying away from the cardiac surgery of the future: virtual patient twins and prediction of the progress of natural tissue wear and tear or risk of injury [3, 24, 37]. It seems that this holistic approach to the topic under

discussion will be of benefit to both engineers and clinicians, allowing them to delve deeper into the subject and develop their own replication of the cited studies or further research.

2. Materials and methods

We provided a careful review of 6 leading medical and technical bibliometric databases (PubMed, EBSCO, EMBASE, PEDro, dblp, and ACM Digital Library) using specified keywords (virtual reality, VR, cardiac surgery, and similar) in accordance with the review methodology shown in Figure 1.

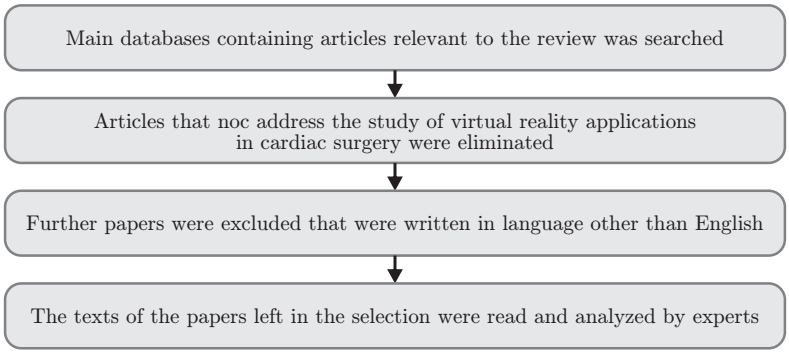


Figure 1. Methodology of the review

Despite the apparent popularity of the topic of VR use in cardiac surgery, only 47 articles published between 2002 and 2022 met the inclusion criteria for the review (Fig. 2).

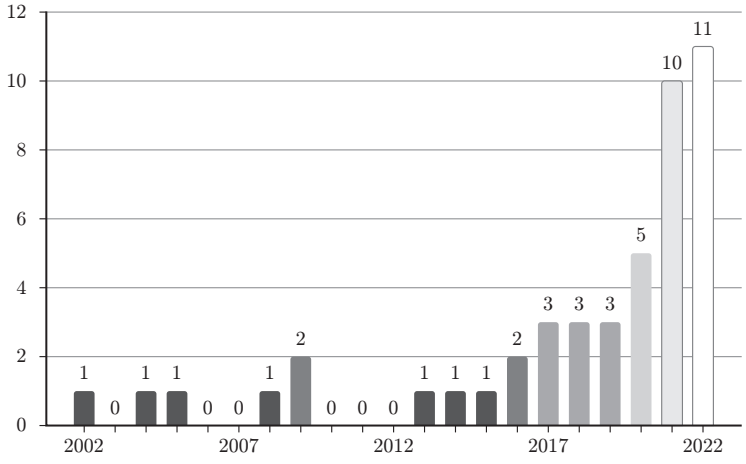


Figure 2. Number of publications selected to the review

Analyzing Figure 2, it can be seen that the number of articles and scientific studies on VR-based solutions in cardiac surgery is constantly growing. This means that the use of modern technologies in cardiac surgery is significant and brings better and better results. Hence the interest of scientists in this subject. Next we will describe the VR methods that are currently used in cardiac surgery based on a review of the available literature on this subject.

3. VR-based methods in medicine

Nowadays the VR processes are being used in an increasing number of daily life activities, fully or partially replacing real processes [19]. Technological developments in the area of VR creation offer opportunities to use this technological advance in many areas of science, economy and social activity. Although VR is a relatively young field, medical science and clinical practice have already learned to benefit from it in the areas of didactics, diagnosis and patient treatment [2, 15]. As far as didactics is concerned, VR allows medical professionals to improve their skills by being able to repeat activities repeatedly with rare clinical cases. At the initial training stage, e.g. for surgeons, it allows a greater margin for errors to be made without affecting the patient. An additional aim of such training is to improve automatic coordination between the monitor and manual procedures. Such an opportunity allows one to focus on the important aspects and not, for example, on what the hands are currently doing [13].

VR has a wide range of applications in neurological rehabilitation – mainly due to the ease of mapping the natural environment, creating specific movement patterns and attractive exercises in which the patient actively participates [16]. Patients learn to use a specific activity in the virtual world and then, with the active supervision of the therapist, transfer specific movement patterns to everyday activities [69]. VR can also support and enhance basic forms of treatment that require repetitive exercises that are tedious and often boring from the patient's point of view. In the case of cardiac patients – those with coronary heart disease – improvements in the mental joint have been demonstrated when classical cardiac rehabilitation is supplemented with VR elements [36].

In the cognitive sphere, virtual reality is gaining an advantage over the classic form of pen-and-paper-based neuropsychological therapy [101]. Virtual reality makes it possible to replicate real-life situations with real-time images of the patient. The course and progress of therapy can also be easily reported, showing the patient the effects of the therapy, which facilitates further diagnosis and modification of the therapy plan by the clinician. Importantly, due to the nature of the therapy provided, virtual reality can be carried out remotely, e.g. in chronic cases where it is difficult for the patient to reach the centre, with simultaneous control of the therapy by the therapist. Additionally, the costs of this form of rehabilitation have been shown to be lower than its traditional form [52].

4. VR-based methods in cardiac surgery

Among reviewed articles only four were review-type articles. Meta-analyses and comprehensive reviews are lacking. Intractable VR allows 3D models of complex intracardiac and extracardiac anatomy to be viewed, even in infants, the ability for users to define their own views based on existing medical diagnostic data. This provides a useful complement to traditional preoperative planning methods, the opportunity for group discussion by the treatment team (including cardiologists and cardiac surgeons), reliability, rapid learning, cost-effectiveness and ease of use [65]. Medical simulation in cardiac surgery, including VR-based simulation, improves trainee learning and performance by allowing repeated training until the required high level of mastery of specific cardiac competencies is achieved, but further research is required on how this translates to the performance of tasks in the operating theatre in relation to a real patient. Cardiac surgery simulation is not yet part of the training program, simulators are available for some tasks and procedures, but based on three different types of simulators [90]:

1. full manikin simulators,
2. partial task trainers,
3. virtual reality systems or combinations of these, including 3D-printed components of a specific patient's cardiovascular system.

There is a strong interest in Head-Mounted Displays (HMDs) and smart glasses in cardiac and vascular surgery for the education of surgeons and for surgical practice, but further technical improvements and clinical trials on large groups are needed for their fuller implementation [47]. Early mobilisation of patients in the post-operative period of cardiac surgery either, in the immediate post-operative period or on the first postoperative day, has positive results. It uses early bedside activity, VR, progressive mobilisation, resistance exercises, cycle ergometer and walking protocols, rather not personalised, low progressive intensity, twice a day for up to 30 minutes [5].

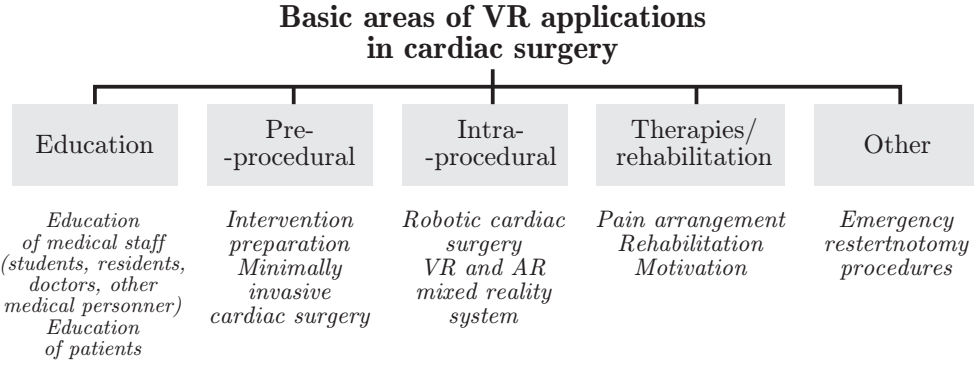


Figure 3. Main areas of VR application in cardiac surgery according to the review

From the point of view of cardiac surgery, VR involves a computer-generated three-dimensional simulation that the user (surgeon, student, their entire team – each from their own perspective) sees and manipulates [9, 64]. Research to date shows that VR improves teaching, but does not completely replace traditional teaching methods [64, 88]. Furthermore, learning with 360° VR video is more effective than learning with 2D video, i.e. realism is key [9]. Within the computer-based teaching module (CBTM), three levels of design were identified [48]:

1. **global level** (goal management, framing, minimising technical load);
2. **rhetorical level** (optimising modalities, making modalities explicit, scaffolding, development, spaced repetition);
3. **specific level** (text management, device management).

Figure 3 shows the main application areas of VR-solutions in cardiac surgery. For each of these scientific areas, a review of the latest articles was carried out, and below we present the state-of-art results.

4.1. VR-supported education of medical professionals

The complexity of cardiac surgery requires ongoing education and training, with an audience of different people and their teams: doctors (cardiologists, cardiac surgeons, anaesthetists), students, as well as patients and their families/carers [6, 8, 18, 66, 74]. Education of medical specialists with the use of VR has its origins in the 90s of the XX century [39], while the beginnings of education in the field of the cardiovascular system using VR date back to the beginning of this century [17, 18]. From the outset, VR has been evaluated as a technology with the potential to support the teaching and assessment of clinical skills of students, residents and doctors. It has fostered the development of medical simulation centres, among others, as it allows various degrees of 'immersion' of trainees in an environment reflecting real clinical situations with such a high degree of accuracy that the clinical skills (diagnostic, therapeutic, etc.) thus acquired are later transferred to patients [73]. The multiplicity of scenarios allow more scenarios (variations in anatomy, pathology, interventions, complications) to be practised in this way than in the real world at the same time (e.g. because there are not enough patients to train). This applies primarily to young resident doctors who should train as much as possible and have contact with various cases, so that in the future they can cope with any situation, even with difficult operations [91]. What's more, the trainees can repeat them in VR many times in different variants without harming the patients until they are completely mastered. Cardiac surgery requires an integrated system for teaching both, from the point of view of content (multimedia elements, their decomposition and description possibilities within teaching scenarios) and information technology within the chosen educational environment. This allows not only the integration in practice of the knowledge and experience acquired during traditional forms of teaching, but also the seamless movement of the trainer across different levels of teaching, student sophistication or even (in medical simulation) the integration of team activities [18]. During COVID-19, the use of simulators for cardiac

surgery allowed for continuous practice of doctors, so that after the pandemic they could go to surgery without unnecessary interruption in training [58]. Interactive VR available locally and via the World Wide Web is used both to teach, analyse and describe cardiac anatomy and to prepare surgical techniques. Their advantages over traditional teaching techniques include a realistic rendering of the sequence of events and spatial considerations during a cardiac surgical procedure [17]. Such solutions are widely accepted. Noorali *et al.* proposed in-house Pakistani solution (simulation lab) providing interdisciplinary VR-based training for promising cardiac surgeons [63].

The use of VR in education is also of utmost importance in the paediatric cardiac intensive care unit [72]. Children, due to their size, are much more difficult cardiac patients and greater precision is required during surgery. Without proper doctor training, performing a cardiac surgery on a child can result in many complications during the procedure. The use of VR-based technology during the doctor's education gives him better training facilities and greater skills to perform such complex operations.

Advances in VR and related devices such as endoscopes and cardiac robots could enable the development of new therapies for severe heart disease. This will require the acquisition of new skills, also on simulators, as has been the case so far, e.g. when learning stimulation and ablation [83].

4.2. VR-supported education for patients and their families

VR can be used as a tool not only to educate doctors and medical staff, but also patients. The use of this technology allows the patient to prepare for surgery and understand the entire process that awaits him. The use of VR increases the understanding, knowledge or comprehension of the patient. An additional advantage of using VR technology is increased satisfaction and reduced anxiety among patients. It also affects the patient's positive perception of the activities performed by the medical personnel [44].

VR becomes an important motivation and learning tool for patients in cardiac surgery [5]. Patients undergoing rehabilitation after cardiac surgery have different needs and preferences. They need a sense of security and seek additional advice. Any e-learning program, including one based on VR, reduces their uncertainty and improves overall mental well-being which supports their faster recovery [27]. Similar VR-based patients' educational programs are studied in obesity and diabetes [12].

4.3. VR-based intervention preparation and minimally invasive cardiac surgery

We decided to separate the planning of the procedure from teaching medical specialists due to its specificity, target group and the importance of benefits that this area of development of VR systems supporting cardiac surgeons may bring, especially in difficult and atypical cases [61, 65]. Successful cardiac surgery procedures require a thorough understanding of the complex anatomy and pathophysiology of

the cardiovascular system. This improves the spatial and temporal understanding of pathological changes and their dynamics when using surgical access and performing the procedure. In a broader context, such an approach accelerates the preparation of procedures, which is important in a large number of urgent cases, when the time to prepare the team is short and the procedure saves the patient's life [41]. It is also an important issue in the case of unusual or complex cases where the use of VR before surgery may modify a surgical procedure [61].

Wierzbicki *et al.* proposed in 2004 Virtual Cardiac Surgery Planning (VCSP) [95] a VR model of the chest made individually for each patient from preoperative medical imaging (CT, MRI). The requirements for such solutions include, above all, adaptability, ease of use and a fairly high degree of accuracy (MSE approx. 1 mm), also in reflecting the dynamics of the operation, for increased reliability in training, planning and conducting cardiac surgery. In addition, limitations due to patients safety, the need for full coverage of the heart with dynamic images from X-rays and angiograms, and the need to use an endoscope to navigate the instruments must be taken into account [68]. In their approach, a static heart model is created by segmenting one of the frames (an image, i.e. a 4D data set), and then based on the remaining frames, the dynamics extracted from the remaining frames of the image is added based on a proprietary algorithm. A similar solution based on CT in robotic cardiac surgery was shown by Ivanov *et al.* in [33]. VR-based integration of CT scans and endoscopic images showed mean spatial error 1.4 mm and time discrepancy in the range of 50–100 ms [82]. The above-mentioned parameters improve with the development of technology.

The 3D cardiographic virtual endoscopy based on MRI and CT scans can be useful in cardiac surgery of children. It achieves diagnostic accuracy ranging from 92.4% to 98.7% [96]. Vigil *et al.* showed in [89] that VR modelling (based on MRI) of the septal pathway and subsequent development of septal templates and visualisation of the access pathway can be beneficial in the preoperative planning of complex double-outlet right ventricle repairs. Ghosh *et al.* in [22] demonstrated the usefulness of VR-based cardiac preparation in a pediatric center with high patient traffic. It uses MRI or CT images to segment the image and transform it into VR with FDA-approved software. Interestingly, in addition to VR, 3D printed models and digital 2D models are also used, with the option of surgical repair made in CAD are designed digitally using proprietary open source computer-aided (CAD) modeling tools. The legitimacy of using the above-mentioned Clinical modeling is shown by statistics: in 3 consecutive years (2018–2021) the demand for it in children has tripled, and in 2020 3, 4 and 5 STAT categories were requested in more than 25% of cases. It is worth noting that the most common indications for modeling in children were complex 2-chamber repair (31%) and repair of multiple defects in the interventricular septum (12%) [22]. In the case of children, the accuracy of the computer model is 0.54 mm and the accuracy of 3D printing is 0.05 mm compared to the digital equivalent [67]. A very interesting option is the transformation of 3D echocardiographic and cardiac

CT data into VR models of higher diagnostic quality, with more accurate measurements and faster navigation [62]. The difference is in the time of obtaining the finished model, where the median time of post-processing VR (DIVA i.e. directly applicable MRI data without intermediate segmentation) was 5 min compared to 8–12 h for 3D printed models [71]. So the time advantage of VR-based models is obvious. The VR systems supporting both preoperative imaging (based on e.g. MRI) and intraoperative in vivo (based on e.g. ultrasound) are more and more often integrated with models of surgical instruments, offering 4.8 mm RMS alignment accuracy [49–51]. The observed difficulties in the interaction of surgeons with the VR environment result from their speed of orientation, insufficient depth information and delegation of view control with an emphasis on the efficiency of the user and his workload [53].

4.4. VR-based robotic cardiac surgery

The last 20 years have brought significant advances in the field of automated minimally invasive cardiac surgery being a safer and more effective solution for some patients from traditional cardiac surgery [33]. In 1998, the da Vinci robotic system was first used for cardiothoracic surgery [34, 94]. Undoubtedly, cardio-surgical robots have increased the capabilities and precision of surgeons, especially in the areas of mitral valve surgery, closure of the atrial septal defect, and direct coronary artery bypass surgery [31, 32].

A study carried out by Chiu *et al.* [10] showed different effectiveness of active involvement of peer observation, in addition to expert demonstration in VR tasks, such as camera control, stratification and switching, energy and seam sponge in da Vinci skill simulators. The effectiveness of such VR observation still requires optimization in order to ensure the best possible learning outcomes. Interestingly, medical students (females) achieved better results in the VR task involving a spongy suture and obtained more stitches, which indicates the need to differentiate training depending on gender [11]. The usefulness of VR-based training in preparation for the use of the da Vinci robot in trainees was also shown by Gleason *et al.* [23].

According to the results of a randomised controlled trial (RCT) conducted by Valdis *et al.* [85, 86], it was proven that the VR helps with cost-effective, high-performance simulation exercises in cardiac robotics.

During the operation, mixed reality systems combining VR and AR are used. An example is the system of Sentiar, Inc., St. Louis, MO deployed on a Microsoft HoloLens (Microsoft Inc., Redmond, WA) [55, 80]. During the operation, the doctor uses a headset, where he has a 3D visualisation of the patient's cardiac system and controls it hands-free. The doctor can get information about the location of the catheter in real time.

4.5. VR-based pain management during cardiac surgery

VR-based pain management in patients after cardiac surgery alleviates vital parameters, reduces discomfort and postoperative stress [60]. But more research is needed

in order to determine if it may occur and what approach to pain and anxiety, for example, in intensive care units [46]. VR also allows you to effectively reduce preoperative anxiety without the use of pharmacological agents [1, 29, 93]. Studies have observed the effect of using VR for postoperative rehabilitation on reducing the use of analgesics in hospitals [57, 84]. VR works well for pain relief, especially after cardiac surgery due to the so-called Gate Theory of Attention. According to this theory, if the patient's attention is diverted and occupied with other activities, he will forget about the pain he is feeling [35]. This mechanism is used in the case of the VR-based tool to reduce pain.

4.6. VR-supported cardiac rehabilitation of patients

There is a widespread belief that VR technologies can provide controlled, multisensory, interactive 3D stimulus environments that engage patients in interventions and measure, record and motivate required human performance. In particular, this can be achieved by promoting desired health behaviours through motivational reinforcement, personalised learning methods and social networks. Moreover, this can be effective even in the case of increasing rates, high prevalence and adverse consequences of disease [12]. VR-based rehabilitation compared to the traditional approach in the control group showed better functional outcomes in patients undergoing cardiac surgery, expressed in outcomes of the Functional Independence Measure (FIM), the 6 minute walk test (6 MWT), and the Nottingham Health Profile (NHP) [4]. To date, post-operative cardiac rehabilitation programs have a number of limitations related both to patients' musculoskeletal problems themselves and more broadly to psychological or existential issues related to lifestyle and health responsibilities [27, 28, 79]. Wider implementation of VR may help to better design such future programs [27].

VR technology can be used during various stages of rehabilitation and many of its features allow to qualify it as a complete rehabilitation tool. In the process of rehabilitation, a very important aspect is the patient's motivation to work on recovery. Tedious exercises can demotivate patients, who therefore adhere less to the recommendations. VR technology increases motivation and makes exercises more interesting and accessible to the patient. Especially in connection with video games [20]. Patients using interactive virtual reality are more active, feel less pain and recover faster after cardiac surgery.

4.7. Other applications of VR in cardiac surgery

VR simulation for purposes of training of cardiopulmonary resuscitation as far as emergency resternotomy procedures after cardiac surgery were developed by Sadeghi *et al.* [76]. The proposed solution is at the proof of concept stage and requires further research . However, the use of VR in cardiac surgery may expand in the coming years to various applications. Virtual reality is also used to create 3D images of cardiac anatomy and VR connects with AG to mixed reality, which creates even more application possibilities [55, 71].

5. Discussion

Overall, the results of our review confirm the opinion on the prospect of research into the applications of VR in cardiac surgery. We found only one article questioning VR support for cardiac surgery. It concerns the 3D diagnosis of congenital atrioventricular valves, yet the authors' doubts do not concern the value of the method itself, but whether its validity is certain at this early stage of development. The authors do not deny that the method itself, when refined, can improve surgeons' understanding of the nature of the defect and help them formulate a repair strategy [59]. Such discussions are desired in science and should take place at such early stages in the development of individual solutions, allowing scientists and engineers to improve them, and clinicians to choose the best one. At the same time, two competing technologies were indicated: Augmented Reality (AR) [70, 75] and 3D printing [38, 42, 97, 99]. This has important implications in terms of the directions of further comparative research between the above-mentioned three main technologies.

As directions for further research, it is crucial to define the differences between teaching using VR (also AR) and traditional quality teaching methods, as well as training methodologies combining/interweaving the above mentioned teaching modalities [7, 64]. Combining multi-modal sensory data and emotion assessment are also available, including analysis and simulation by artificial intelligence [7]. A very important direction of further research is the recognition of clinical needs – the coordination of knowledge and experience of engineers, scientists and clinicians may lead to the development of new fields of VR applications in previously unexplored areas of cardiac surgery, which we may not even know are within reach (e.g. cardiac surgery), preventive, micro- or nanorobotics, cardiosurgical neuroprosthetics based on bioMEMS and bioNEMS). Clinical 3D modelling must be integrated with the pre-operative care of patients with heart defects, and the demand for these services is growing rapidly [22, 43].The use of VR in cardiac surgery is not just in one area and the benefits may be subject to synergistic effects. Further work is needed to fully demonstrate the clinical benefits and improved outcomes in post-cardiac surgery patients as a result of VR-based methods and/or tools.

An interesting area of VR applications is prehabilitation [14, 30], i.e. preparing patients for cardiac surgery. Each such procedure, no matter how minimally invasive, is a challenge for the body. Hence the body needs to be properly prepared for it: physically (diet, activity) and mentally (positive attitudes, motivation to change your lifestyle to a healthier one as part of medicine). Hirota proposed that VR-prehabilitation may be a promising tool for the prevention of postoperative delirium POD [30]. This will turn cardiac interventions into a type of personalised targeted therapy to avoid relapse.

The VR/AR-based curriculum platform should be standardised to benchmark the development of basic robotic skills, provide common interdisciplinary surgical education and objectify student achievement. There is also a lack of studies and publications on the standardisation of management in AV/AR-assisted cardiac surgery.

Further large-scale randomised clinical trials on large homogeneous groups of patients are needed to develop standards. These will increase as the technology itself becomes more widespread.

6. Conclusions

The application of VR technology to cardiac surgery has many aspects and development possibilities. Taking into account the current advancement of work in this field, it can be expected that in the near future, training of doctors and residents will be based on VR-based virtual and real cases. Cardiac surgery simulation should be a part of the training program. Combining VR with MRI and CT imaging can also create models of hearts in difficult cases on which the surgeon can practise before starting surgery on the patient. Another aspect in the VR-based support of cardiac surgery will be the development of cardiac robots using VR to perform semi-automatic or automatic surgery. The next step will be carrying out remote surgery using VR imaging. The specialist will not have to be physically in the same place as the patient to participate in cardiac surgery. Using VR and robots, the doctor will see what is happening during the operation, on this basis, will be able to decide on the next steps of the operation and will perform the surgery procedure using this robot. While pre- and post-operative VR-based solutions help patients with rehabilitation and pain management. This approach will certainly influence the patients' well-being and faster return to health.

Currently, VR and AR are used in all the above-mentioned aspects of cardiac surgery. In combination with Industry 4.0 technologies, the health care system is improved and Healthcare 4.0 systems are created. Such systems take advantage of various technologies to improve patient health care. Currently, both VR and AR applications are being considered. Both technologies complement each other and in the near future there will probably be solutions based on one or the other technology implemented in care systems during cardiac surgery.

To sum up the field of VR/AR applications in cardiac surgery is growing rapidly, and researchers, medical professionals and technology developers continue to explore innovative ways to use immersive technologies to improve surgical outcomes and patient care:

1. surgical training and simulation: enhanced training modules and integration of haptic feedback,
2. patient-specific models based on medical imaging data for pre-operative planning with AR overlay in surgery (showing 3D reconstructions, vital signs or navigation cues and other data,
3. telemedicine and remote support: remote consultations (providing real-time guidance and support) and training support in remote areas with limited access to medical expertise,
4. data integration and analysis and visualisation in a convenient form for the surgeon/team, as well as decision support systems.

VR-based solutions in cardiac surgery are useful both for medical specialists at various levels of professional development and for the patients themselves. The VR-based curriculum platform should be standardised to compare the development of basic robotic skills, provide a common interdisciplinary surgical education, and objectify student achievement. The new lifestyle required from cardiac surgery patients is easier to implement thanks to VR-based educational and motivational tools. However, it is necessary to develop the above-mentioned tools and compare their effectiveness with AR. With the aforementioned reasons, interdisciplinary collaboration between scientists, clinicians and engineers is necessary.

References

[1] Aardoom J.J., Hilt A.D., Woudenberg T., Chavannes N.H., Atsma D.E.: A Preoperative Virtual Reality App for Patients Scheduled for Cardiac Catheterization: Pre-Post Questionnaire Study Examining Feasibility, Usability, and Acceptability, *JMIR Cardio*, vol. 6(1), e29473, 2022. doi: 10.2196/29473.

[2] Alfalah S.F.M., Falah J., Alfalah T., Elfalah M., Muhaidat N., Falah O.: A comparative study between a virtual reality heart anatomy system and traditional medical teaching modalities, *Virtual Reality*, vol. 23, pp. 229–234, 2019. doi: 10.1007/s10055-018-0359-y.

[3] Alonzo M., AnilKumar S., Roman B., Tasnim N., Joddar B.: 3D Bioprinting of cardiac tissue and cardiac stem cell therapy, *Translational Research*, vol. 211, pp. 64–83, 2019. doi: 10.1016/j.trsl.2019.04.004.

[4] Assis Pereira Cacau de L., Uruga Oliveira G., Godinho Maynard L., Araújo Filho de A.A., Monteiro da Silva Jr W., Cerqueria Neto M.L., Antoniolli A.R., Santana-Filho V.J.: The use of the virtual reality as intervention tool in the postoperative of cardiac surgery, *Revista Brasileira de Cirurgia Cardiovascular*, vol. 28(2), pp. 281–289, 2013. doi: 10.5935/1678-9741.20130039.

[5] Barbosa Borges M.G., Lago Borges D., Oliveira Ribeiro M., Silva Lima L.S., Carneiro Morais Macedo K., da Silva Nina V.J.: Early Mobilization Prescription in Patients Undergoing Cardiac Surgery: Systematic Review, *Brazilian Journal of Cardiovascular Surgery*, vol. 37(2), pp. 227–238, 2022. doi: 10.21470/1678-9741-2021-0140.

[6] Borger M.: The future of cardiac surgery training: A survival guide, *Journal of Thoracic and Cardiovascular Surgery*, vol. 154(3), pp. 994–995, 2017. doi: 10.1016/j.jtcvs.2017.04.060.

[7] Bălan O., Moise G., Moldoveanu A., Leordeanu M., Moldoveanu F.: An Investigation of Various Machine and Deep Learning Techniques Applied in Automatic Fear Level Detection and Acrophobia Virtual Therapy, *Sensors*, vol. 20, 496, 2020. doi: 10.3390/s20020496.

[8] Chakravarthy M.: Future of awake cardiac surgery, *Journal of Cardiothoracic and Vascular Anesthesia*, vol. 28(3), pp. 771–777, 2014. doi: 10.1053/j.jvca.2013.03.005.

[9] Chao Y.P., Chuang H.-H., Hsin L.-J., Kang C.-J., Fang T.-J., Li H.-Y., Huang C.-G., *et al.*: Using a 360° Virtual Reality or 2D Video to Learn History Taking and Physical Examination Skills for Undergraduate Medical Students: Pilot Randomized Controlled Trial, *JMIR Serious Games*, vol. 9(4), e13124, 2021. doi: 10.2196/13124.

[10] Chiu H., Kang Y., Wang W., Chen C., Hsu W., Tseng M., Wei P.: The Role of Active Engagement of Peer Observation in the Acquisition of Surgical Skills in Virtual Reality Tasks for Novices, *Journal of Surgical Education*, vol. 76(6), pp. 1655–1662, 2019. doi: 10.1016/j.jsurg.2019.05.004.

[11] Chiu H.-Y., Kang Y.-N., Wang W.-L., Tong Y.-S., Chang S.-W., Fong T.-H., Wei P.-L.: Gender differences in the acquisition of suturing skills with the da Vinci surgical system, *Journal of the Formosan Medical Association*, vol. 119(1 Part 3), pp. 462–470, 2020. doi: 10.1016/j.jfma.2019.06.013.

[12] Ershow A., Peterson C., Riley W., Rizzo A., Wansink B.: Virtual reality technologies for research and education in obesity and diabetes: research needs and opportunities, *Journal of Diabetes Science and Technology*, vol. 5(2), pp. 212–224, 2011. doi: 10.1177/193229681100500202.

[13] Eysenck M., Keane M.: Attention and performance limitations. In: D.J. Levitin (ed.), *Foundations of Cognitive Psychology: Core Readings*, pp. 363–398, MIT Press, Cambridge, MA, 2002.

[14] Fernández-Costa D., Gómez-Salgado J., del Río A.C., Borrallo-Riego A., Guerra-Martin M.D.: Effects of prehabilitation on functional capacity in aged patients undergoing cardiothoracic surgeries: a systematic review, *Healthcare*, vol. 9(11), 1602, 2021. doi: 10.3390/healthcare9111602.

[15] Fidurski K., Falkowski-Gilski P.: Nauka w świecie cyfrowym okiem młodego inżyniera – początki techniki wirtualnej rzeczywistości, *Pismo PG*, vol. 1, pp. 30–32, 2022.

[16] Fluet G., Deutsch J.: Virtual Reality for Sensorimotor Rehabilitation Post-Stroke: The Promise and Current State of the Field, *Current Physical Medicine and Rehabilitation Reports*, vol. 1(1), pp. 9–20, 2013. doi: 10.1007/s40141-013-0005-2.

[17] Friedl R., Preisack M.B., Klas W., Rose T., Stracke S., Quast K.J., Hannekum A., Gödje O.: Virtual reality and 3D visualizations in heart surgery education, *Heart Surgery Forum*, vol. 5(3), pp. E17–E21, 2002.

[18] Friedl R., Preisack M., Schefer M., Klas W., Tremper J., Rose T., Bay J., *et al.*: CardioOP: an integrated approach to teleteaching in cardiac surgery, *Studies in Health Technology and Informatics*, vol. 70, pp. 76–82, 2000.

[19] Furmanek W., Piecuch A. (eds.): *Dydaktyka informatyki: modelowanie i symulacje komputerowe*, Wydawnictwo Uniwersytetu Rzeszowskiego, Rzeszów, 2010.

[20] García-Bravo S., Cuesta-Gêmez A., Campuzano-Ruiz R., Lêpez-Navas M., Domínguez-Paniagua J., Araujo-Narvçez A., Barreìada-Copete E., *et al.*: Virtual reality and video games in cardiac rehabilitation programs. A systematic review, *Disability and Rehabilitation*, vol. 43(4), pp. 448–457, 2021. doi: 10.1080/09638288.2019.16318.

[21] Gendia A., Rehman M., Cota A., Gilbert J., Clark J.: Can virtual reality technology be considered as a part of the surgical care pathway?, *The Annals of the Royal College of Surgeons of England*, vol. 105(1), pp. 2–6, 2023. doi: 10.1308/rcsann.2022.0125.

[22] Ghosh M.G., Jolley M.A., Mascio C.E., Chen J.M., Fuller S., Rome J.J., Silvestro E., *et al.*: Clinical 3D modeling to guide pediatric cardiothoracic surgery and intervention using 3D printed anatomic models, computer aided design and virtual reality, *3D Printing in Medicine*, vol. 8(1), 11, 2022. doi: 10.1186/s41205-022-00137-9.

[23] Gleason A., Servais E., Quadri S., Manganiello M., Cheah Y.L., Simon C.J., Preston E., *et al.*: Developing basic robotic skills using virtual reality simulation and automated assessment tools: a multidisciplinary robotic virtual reality-based curriculum using the Da Vinci Skills Simulator and tracking progress with the Intuitive Learning platform, *Journal of Robotic Surgery*, vol. 16(6), pp. 1313–1319, 2022. doi: 10.1007/s11701-021-01363-9.

[24] Gokce C., Gurcan C., Delogu L., Yilmazer A.: 2D materials for cardiac tissue repair and regeneration, *Frontiers in Cardiovascular Medicine*, vol. 9, 802551, 2022. doi: 10.3389/fcvm.2022.802551.

[25] Gooding P., Clifford D.: Semi-automated care: Video-algorithmic patient monitoring and surveillance in care settings, *Journal of Bioethical Inquiry*, vol. 18(4), pp. 541–546, 2021. doi: 10.1007/s11673-021-10139-7.

[26] Grab M., Hopfner C., Gesenhues A., König F., Haas N.A., Hagl C., Curta A., Thierfelder N.: Development and evaluation of 3D-printed cardiovascular phantoms for interventional planning and training, *JoVE (Journal of Visualized Experiments)*, vol. 167, e62063, 2021. doi: 10.3791/62063-v.

[27] Hansen T.B., Berg S.K., Sibilitz K.L., Zwisler A.D., Norekvål T.M., Lee A., Buus N.: Patient perceptions of experience with cardiac rehabilitation after isolated heart valve surgery, *European Journal of Cardiovascular Nursing*, vol. 17(1), pp. 45–53, 2018. doi: 10.1177/1474515117716245.

[28] Hansen T.B., Zwisler A.D., Berg S.K., Sibilitz K.L., Buus N., Lee A.: Cardiac rehabilitation patients' perspectives on the recovery following heart valve surgery: a narrative analysis, *Journal of Advanced Nursing*, vol. 72(5), pp. 1097–1108, 2016. doi: 10.1111/jan.12904.

[29] Hendricks T.M., Gutierrez C.N., Stulak J.M., Dearani J.A., Miller J.D.: The Use of Virtual Reality to Reduce Preoperative Anxiety in First-Time Sternotomy Patients: A Randomized Controlled Pilot Trial, *Mayo Clinic Proceedings*, vol. 95(6), pp. 1148–1157, 2020. doi: 10.1016/j.mayocp.2020.02.032.

[30] Hirota K.: Preoperative management and postoperative delirium: the possibility of neuroprehabilitation using virtual reality, *Journal of Anesthesia*, vol. 34(1), pp. 1–4, 2020. doi: 10.1007/s00540-019-02660-2.

[31] Ishikawa N., Watanabe G.: Robot-assisted cardiac surgery, *Annals of Thoracic and Cardiovascular Surgery*, vol. 21(4), pp. 322–328, 2015. doi: 10.5761/atcs.ra.15-00145.

[32] Ishikawa N., Watanabe G.: Ultra-minimally invasive cardiac surgery: robotic surgery and awake CABG, *Surgery Today Official Journal of the Japan Surgical Society*, vol. 45(1), pp. 1–7, 2015. doi: 10.1007/s00595-014-0961-x.

[33] Ivanov N.A., Green D.B., Guy T.S.: Integrate imaging approach for minimally invasive and robotic procedures, *Journal of Thoracic Disease*, vol. 9(Suppl4), pp. S264–S270, 2017. doi: 10.21037/jtd.2017.03.141.

[34] Jin Z.: Clinical application of Da Vinci surgical system in China, *Zhongguo Yi Liao Qi Xie Za Zhi* (*Chinese Journal of Medical Instrumentation*), vol. 38(1), pp. 47–49, 2014.

[35] Jones T., Moore T., Choo J.: The impact of virtual reality on chronic pain, *PLoS ONE*, vol. 11(12), e0167523, 2016. doi: 10.1371/journal.pone.0167523.

[36] Jóźwik S., Wrzeciono A., Cieślik B., Kiper P., Szczepańska-Gieracha J., Gajda R.: The Use of Virtual Therapy in Cardiac Rehabilitation of Male Patients with Coronary Heart Disease: A Randomized Pilot Study, *Healthcare*, vol. 10(4), 745, 2022. doi: 10.3390/healthcare10040745.

[37] Kamel Boulos M., Zhang P.: Digital twins: from personalised medicine to precision public health, *Journal of Personalized Medicine*, vol. 11(8), 745, 2021. doi: 10.3390/jpm11080745.

[38] Kappanayil M., Koneti N.R., Kannan R.R., Kottayil B., Kumar K.: Three-dimensional-printed cardiac prototypes aid surgical decision-making and preoperative planning in selected cases of complex congenital heart diseases: Early experience and proof of concept in a resource-limited environment, *Annals of Pediatric Cardiology*, vol. 10(2), pp. 117–125, 2017. doi: 10.4103/apc.apc_149_16.

[39] Kaufman D.M., Bell W.: Teaching and assessing clinical skills using virtual reality, *Studies in Health Technology and Informatics*, vol. 39, pp. 467–472, 1997.

[40] Kennedy C.W., Hu T., Desai J.P., Wechsler A.S., Kresh J.Y.: A novel approach to robotic cardiac surgery using haptics and vision, *Cardiovascular Engineering: An International Journal*, vol. 2, pp. 15–22, 2002. doi: 10.1023/A:1019926620096.

[41] Kim B., Nguyen P., Loke Y.H., Cleveland V., Liu X., Mass P., Hibino N., *et al.*: Virtual Reality Cardiac Surgical Planning Software (CorFix) for Designing Patient-Specific Vascular Grafts: Development and Pilot Usability Study, *JMIR Cardio*, vol. 6(1), e35488, 2022. doi: 10.2196/35488.

[42] Kiraly L., Shah N.C., Abdullah O., Al-Ketan O., Rowshan R.: Three-Dimensional Virtual and Printed Prototypes in Complex Congenital and Pediatric Cardiac Surgery – A Multidisciplinary Team-Learning Experience, *Biomolecules*, vol. 11(11), 1703, 2021. doi: 10.3390/biom11111703.

[43] Krasemann T., Branstetter J.: Virtual Reality Treatment Planning for Congenital Heart Disease, *JACC Case Reports*, vol. 3(14), pp. 1584–1585, 2021. doi: 10.1016/j.jaccas.2021.08.023.

[44] Kruk van der S.R., Zielinski R., MacDougall H., Hughes-Barton D., Gunn K.M.: Virtual reality as a patient education tool in healthcare: A scoping review, *Patient Education and Counseling*, vol. 105(7), pp. 1928–1942, 2022. doi: 10.1016/j.pec.2022.02.005.

[45] Kumar A., Krishnamurthi R., Nayyar A., Sharma K., Grover V., Hossain E.: A novel smart healthcare design, simulation, and implementation using healthcare 4.0 processes, *IEEE Access*, vol. 8, pp. 118433–118471, 2020. doi: 10.1109/access.2020.3004790.

[46] Laghlam D., Naudin C., Coroyer L., Aidan V., Malvy J., Rahoual G., Estagnasié P., Squara P.: Virtual reality vs. Kalinox© for management of pain in intensive care unit after cardiac surgery: a randomized study, *Annals of Intensive Care*, vol. 11(1), 74, 2021. doi: 10.1186/s13613-021-00866-w.

[47] Lareyre F., Chaudhuri A., Adam C., Carrier M., Mialhe C., Raffort J.: Applications of Head-Mounted Displays and Smart Glasses in Vascular Surgery, *Annals of Vascular Surgery*, vol. 75, pp. 497–512, 2021. doi: 10.1016/j.avsg.2021.02.033.

[48] Lau K.H.V.: Computer-based teaching module design: principles derived from learning theories, *Medical Education in Review*, vol. 48(3), pp. 247–254, 2014. doi: 10.1111/medu.12357.

[49] Linte C.A., Moore J., Wedlake C., Bainbridge D., Guiraudon G.M., Jones D.L., Peters T.M.: Inside the beating heart: an in vivo feasibility study on fusing pre- and intra-operative imaging for minimally invasive therapy, *International Journal of Computer Assisted Radiology and Surgery*, vol. 4(2), pp. 113–123, 2009.

[50] Linte C.A., Moore J., Wiles A.D., Wedlake C., Peters T.M.: Virtual reality-enhanced ultrasound guidance: a novel technique for intracardiac interventions, *Computer Aided Surgery*, vol. 13(2), pp. 82–94, 2008. doi: 10.1080/10929080801951160.

[51] Linte C.A., White J., Eagleson R., Guiraudon G.M., Peters T.M.: Virtual and augmented medical imaging environments: enabling technology for minimally invasive cardiac interventional guidance, *IEEE Reviews in Biomedical Engineering*, vol. 3, pp. 25–47, 2010. doi: 10.1109/rbme.2010.2082522.

[52] Lloréns R., Noé E., Colomer C., Alcañiz M.: Effectiveness, usability, and cost-benefit of a virtual reality-based telerehabilitation program for balance recovery after stroke. A randomized controlled trial, *Archives of Physical Medicine and Rehabilitation*, vol. 96(3), pp. 418–425, 2015. doi: 10.1016/j.apmr.2014.10.019.

[53] Lo J., Moore J., Wedlake C., Guiraudon G., Eagleson R., Peters T.: Surgeon-controlled visualization techniques for virtual reality-guided cardiac surgery, *Studies in Health Technology and Informatics*, vol. 142, pp. 162–167, 2009.

[54] Maciołek J., Wąsek W., Kamiński B., Piotrowicz K., Krzesiński P.: The impact of mobile virtual reality–enhanced relaxation training on anxiety levels in patients undergoing cardiac rehabilitation, *Kardiologia Polska* (*Polish Heart Journal*), vol. 78(10), pp. 1032–1034, 2020. doi: 10.33963/kp.15528.

[55] Maresky H.S., Oikonomou A., Ali I., Ditkofsky N., Pakkal M., Ballyk B.: Virtual reality and cardiac anatomy: Exploring immersive three-dimensional cardiac imaging, a pilot study in undergraduate medical anatomy education, *Clinical Anatomy*, vol. 32(2), pp. 238–243, 2019. doi: 10.1002/ca.23292.

[56] Maynard L.G., de Menezes D.L., Lião N.S., de Jesus E.M., Andrade N.L.S., Santos J.C.D., da Silva Júnior W.M., *et al.*: Effects of exercise training combined with virtual reality in functionality and health-related quality of life of patients on hemodialysis, *Games for Health Journal*, vol. 8(5), pp. 339–348, 2019. doi: 10.1089/g4h.2018.0066.

[57] McFarland M., Zelaya N., Hossain G., Hicks D., McLauchlan L.: Pain Mitigation Through Virtual Reality Applications. In: *2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR)*, pp. A2–5, IEEE, 2019. doi: 10.1109/ismcr47492.2019.8955702.

[58] McKechnie T., Levin M., Zhou K., Freedman B., Palter V.N., Grantcharov T.P.: Virtual surgical training during COVID-19: operating room simulation platforms accessible from home, *Annals of Surgery*, vol. 272(2), e153, 2020. doi: 10.1097/sla.0000000000003999.

[59] Moscatiello M., Lo Rito M.: Commentary: Virtual reality 3-dimensional imaging of atrioventricular valves: A tool for surgeons or a toy for engineers?, *JTCVS Techniques*, vol. 7, pp. 278–279, 2021. doi: 10.1016/j.xjtc.2021.03.021.

[60] Mosso-Vázquez J.L., Gao K., Wiederhold B.K., Wiederhold M.D.: Virtual reality for pain management in cardiac surgery, *Cyberpsychology, Behavior, and Social Networking*, vol. 17(6), pp. 371–378, 2014. doi: 10.1089/cyber.2014.0198.

[61] Napa S., Moore M., Bardyn T.: Advancing cardiac surgery case planning and case review conferences using virtual reality in medical libraries: evaluation of the usability of two virtual reality apps, *JMIR Human Factors*, vol. 6(1), e12008, 2019. doi: 10.2196/12008.

[62] Narang A., Hitschrich N., Mor-Avi V., Schreckenberg M., Schummers G., Tiemann K., Hitschrich D., *et al.*: Virtual Reality Analysis of Three-Dimensional Echocardiographic and Cardiac Computed Tomographic Data Sets, *Journal of the American Society of Echocardiography*, vol. 33(11), pp. 1306–1315, 2020. doi: 10.1016/j.echo.2020.06.018.

[63] Noorali A.A., Hussain Merchant A.A., Babar Chauhan S.S., Khan M.A., Ehsan A.N., Pervez M.B., Tariq M., Fatimi S.H.: Conceptual framework for a cardiac surgery simulation laboratory and competency-based curriculum in Pakistan – a short innovation report, *Journal of the Pakistan Medical Association*, vol. 72(Suppl. 1), pp. S103–S105, 2022. doi: 10.47391/jpma.aku-21.

[64] Ojala S., Sirola J., Nykopp T., Kröger H., Nuutinen H.: The impact of teacher's presence on learning basic surgical tasks with virtual reality headset among medical students, *Medical Education Online*, vol. 27(1), 2050345, 2022. doi: 10.1080/10872981.2022.2050345.

[65] Ong C.S., Krishnan A., Huang C.Y., Spevak P., Vricella L., Hibino N., Garcia J.R., Gaur L.: Role of virtual reality in congenital heart disease, *Congenital Heart Disease*, vol. 13(3), pp. 357–361, 2018. doi: 10.1111/chd.12587.

[66] Pelletier M.P., Kaneko T., Peterson M.D., Thourani V.H.: From sutures to wires: The evolving necessities of cardiac surgery training, *Journal of Thoracic and Cardiovascular Surgery*, vol. 154(3), pp. 990–993, 2017. doi: 10.1016/j.jtcvs.2017.03.157.

[67] Perens G., Chyu J., McHenry K., Yoshida T., Finn J.P.: Three-Dimensional Congenital Heart Models Created With Free Software and a Desktop Printer: Assessment of Accuracy, Technical Aspects, and Clinical Use, *World Journal for Pediatric and Congenital Heart Surgery*, vol. 11(6), pp. 797–801, 2020. doi: 10.1177/2150135120952072.

[68] Peters T.M., Linte C.A., Moore J., Bainbridge D., Jones D.L., Guiraudon G.M.: Towards a medical virtual reality environment for minimally invasive cardiac surgery. In: T. Dohi, I. Sakuma, H. Liao (eds.), *Medical Imaging and Augmented Reality. MIAR 2008. 4th International Workshop Tokyo, Japan, August 1–2, 2008 Proceedings*, Lecture Notes in Computer Science, vol. 5128, Springer, Berlin–Heidelberg. doi: 10.1007/978-3-540-79982-5_1.

[69] Proffitt R., Lange B.: Considerations in the efficacy and effectiveness of virtual reality interventions for stroke rehabilitation: moving the field forward, *Physical Therapy*, vol. 95(3), pp. 441–448, 2015. doi: 10.2522/ptj.20130571.

[70] Rad A.A., Vardanyan R., Lopuszko A., Alt C., Stoffels I., Schmack B., Ruhparwar A., *et al.*: Virtual and Augmented Reality in Cardiac Surgery, *Brazilian Journal of Cardiovascular Surgery*, vol. 37(1), pp. 123–127, 2022. doi: 10.21470/1678-9741-2020-0511.

[71] Raimondi F., Vida V., Godard C., Bertelli F., Reffo E., Boddaert N., El Beheiry M., Masson J.: Fast-track virtual reality for cardiac imaging in congenital heart disease, *Journal of Cardiac Surgery*, vol. 36(7), pp. 2598–2602, 2021. doi: 10.1111/jocs.15508.

[72] Ralston B.H., Willett R.C., Namperumal S., Brown N.M., Walsh H., Muñoz, R.A., Del Castillo S., *et al.*: Use of virtual reality for pediatric cardiac critical care simulation, *Cureus*, vol. 13(6), e15856, 2021. doi: 10.7759/cureus.15856.

[73] Ramphal P.S., Coore D.N., Craven M.P., Forbes N.F., Newman S.M., Coye A.A., Little S.G., Silvera B.C.: A high fidelity tissue-based cardiac surgical simulator, *European Journal of Cardio-Thoracic Surgery*, vol. 27(5), pp. 910–916, 2005. doi: 10.1016/j.ejcts.2004.12.049.

[74] Reardon M.J.: Change is the only constant, *Journal of Thoracic and Cardiovascular Surgery*, vol. 154(3), pp. 996–997, 2017. doi: 10.1016/j.jtcvs.2017.03.100.

[75] Sadeghi A.H., El Mathari S., Abjigitova D., Maat A.P.W.M., Taverne Y.J.H.J., Bogers A.J.C., Mahtab E.A.F.: Current and Future Applications of Virtual, Augmented, and Mixed Reality in Cardiothoracic Surgery, *Annals of Thoracic Surgery*, vol. 113(2), pp. 681–691, 2022. doi: 10.1016/j.athoracsur.2020.11.030.

[76] Sadeghi A.H., Peek J.J., Max S.A., Smit L.L., Martina B.G., Rosalia R.A., Bakhuis W., *et al.*: Virtual Reality Simulation Training for Cardiopulmonary Resuscitation After Cardiac Surgery: Face and Content Validity Study, *JMIR Serious Games*, vol. 10(1), e30456, 2022. doi: 10.2196/30456.

[77] Sanders J., Bowden T., Woolfe-Loftus N., Sekhon M., Aitken L.: Predictors of health-related quality of life after cardiac surgery: a systematic review, *Health and Quality of Life Outcomes*, vol. 20(1), pp. 1–12, 2022. doi: 10.1186/s12955-022-01980-4.

[78] Sharma R., Singh D., Gaur P., Joshi D.: Intelligent automated drug administration and therapy: future of healthcare, *Drug Delivery and Translational Research*, vol. 11, pp. 1878–1902, 2021. doi: 10.1007/s13346-020-00876-4.

[79] Sibilitz K.L., Berg S.K., Rasmussen T.B., Risom S.S., Thygesen L.C., Tang L., Hansen T.B., *et al.*: Cardiac rehabilitation increases physical capacity but not mental health after heart valve surgery: a randomised clinical trial, *Heart*, vol. 102(24), pp. 1995–2003, 2016. doi: 10.1136/heartjnl-2016-309414.

[80] Silva J., Southworth M., Raptis C., Silva J.: Emerging applications of virtual reality in cardiovascular medicine, *JACC: Basic to Translational Science*, vol. 3(3), pp. 420–430, 2018. doi: 10.1016/j.jacbts.2017.11.009.

[81] Skalidis I., Muller O., Fournier S.: CardioVerse: The cardiovascular medicine in the era of Metaverse, *Trends in Cardiovascular Medicine*, vol. 33(8), pp. 471–476. doi: 10.1016/j.tcm.2022.05.004.

[82] Szpala S., Wierzbicki M., Guiraudon G., Peters T.M.: Real-time fusion of endoscopic views with dynamic 3-D cardiac images: a phantom study, *IEEE Transactions on Medical Imaging*, vol. 24(9), pp. 1207–1215, 2005. doi: 10.1109/tmi.2005.853639.

[83] Talbot H., Spadoni F., Duriez C., Sermesant M., O'Neill M., Jaïs P., Cotin S., Delingette H.: Interactive training system for interventional electrocardiology procedures, *Medical Image Analysis*, vol. 35, pp. 225–237, 2017. doi: 10.1016/j.media.2016.06.040.

[84] Theingi S., Leopold I., Ola T., Cohen G.S., Maresky H.S.: Virtual Reality as a Non-Pharmacological Adjunct to Reduce the Use of Analgesics in Hospitals, *Journal of Cognitive Enhancement*, vol. 6, pp. 108–113, 2022. doi: 10.1007/s41465-021-00212-9.

[85] Valdis M., Chu M.W.A., Schlachta C.M., Kiaii B.: Validation of a Novel Virtual Reality Training Curriculum for Robotic Cardiac Surgery: A Randomized Trial, *Innovations*, vol. 10(6), pp. 383–388, 2015. doi: 10.1097/imi.0000000000000222.

[86] Valdis M., Chu M.W.A., Schlachta C.M., Kiaii B.: Evaluation of robotic cardiac surgery simulation training: A randomized controlled trial, *Journal of Thoracic and Cardiovascular Surgery*, vol. 151(6), pp. 1498–1505.e2, 2016. doi: 10.1016/j.jtcvs.2016.02.016.

[87] Venkatesan M., Mohan H., Ryan J.R., Schurch C.M., Nolan G.P., Frakes D.H., Coskun A.F.: Virtual and augmented reality for biomedical applications, *Cell Reports Medicine*, vol. 2(7), 100348, 2021. doi: 10.1016/j.xcrm.2021.100348.

[88] Vervoort D., Fiedler A.G.: Virtual reality, e-learning, and global cardiac surgical capacity-building, *Journal of Cardiac Surgery*, vol. 36(6), pp. 1835–1837, 2021. doi: 10.1111/jocs.15498.

[89] Vigil C., Lasso A., Ghosh R.M., Pinter C., Cianciulli A., Nam H.H., *et al.*: Modeling Tool for Rapid Virtual Planning of the Intracardiac Baffle in Double-Outlet Right Ventricle, *Annals of Thoracic Surgery*, vol. 111(6), pp. 2078–2083, 2021. doi: 10.1016/j.athoracsur.2021.02.058.

[90] Villanueva C., Xiong J., Rajput S.: Simulation-based surgical education in cardiothoracic training, *ANZ Journal of Surgery*, vol. 90(6), pp. 978–983, 2020. doi: 10.1111/ans.15593.

[91] Vinck E.E., Smood B., Barros L., Palmen M.: Robotic cardiac surgery training during residency: Preparing residents for the inevitable future, *Laparoscopic, Endoscopic and Robotic Surgery*, vol. 5(2), pp. 75–77, 2022. doi: 10.1016/j.lers.2022.03.002.

[92] Wang C., Zhang L., Qin T., Xi Z., Sun L., Wu H., Li D.: 3D printing in adult cardiovascular surgery and interventions: a systematic review, *Journal of Thoracic Disease*, vol. 12(6), 3227, 2020. doi: 10.21037/jtd-20-455.

[93] Wang L., Liu J., Xie W., Chen Q., Cao H.: Condition notification assisted by virtual reality technology reduces the anxiety levels of parents of children with simple CHD: a prospective randomised controlled study, *Cardiology in the Young*, vol. 32(11), pp. 1801–1806, 2022. doi: 10.1017/S104795112100500X.

[94] Watanabe G., Ishikawa N.: da Vinci surgical system, *Kyobu Geka* (*Japanese Journal of Thoracic Surgery*), vol. 67(8), pp. 686–689, 2014.

[95] Wierzbicki M., Drangova M., Guiraudon G., Peters T.: Validation of dynamic heart models obtained using non-linear registration for virtual reality training, planning, and guidance of minimally invasive cardiac surgeries, *Medical Image Analysis*, vol. 8(3), pp. 387–401, 2004. doi: 10.1016/j.media.2004.06.014.

[96] Xue H., Sun K., Yu J., Chen B., Chen G., Hong W., Yao L., Wu L.: Three-dimensional echocardiographic virtual endoscopy for the diagnosis of congenital heart disease in children, *International Journal of Cardiovascular Imaging*, vol. 28(6), pp. 851–859, 2010. doi: 10.1007/s10554-010-9649-5.

[97] Yamada T., Osako M., Uchimuro T., Yoon R., Morikawa T., Sugimoto M., Suda H., Shimizu H.: Three-Dimensional Printing of Life-Like Models for Simulation and Training of Minimally Invasive Cardiac Surgery, *Innovations*, vol. 12(6), pp. 459–465, 2017. doi: 10.1177/155698451701200615.

[98] Yeh L.R., Chen W.C., Chan H.Y., Lu N.H., Wang C.Y., Twan W.H., Du W.C., *et al.*: Integrating ECG monitoring and classification via IoT and deep neural networks, *Biosensors*, vol. 11(6), 188, 2021. doi: 10.3390/bios11060188.

[99] Yoo S., Hussein N., Peel B., Coles J., Arsdell G., Honjo O., Haller C., Lam C., Seed M., Barron D.: 3D Modeling and Printing in Congenital Heart Surgery: Entering the Stage of Maturation, *Frontiers in Pediatrics*, vol. 9, 621672, 2021. doi: 10.3389/fped.2021.621672.

[100] Zanatta F., Farhane-Medina N., Adorni R., Steca P., Giardini A., D'Addario M., Pierobon A.: Combining robot-assisted therapy with virtual reality or using it alone? A systematic review on health-related quality of life in neurological patients, *Health and Quality of Life Outcomes*, vol. 21(1), 18, 2023. doi: 10.1186/s12955-023-02097-y.

[101] Zell E., Dyck E., Kohsik A., Grewe P., Flentge D., Winter Y., Piefke M., *et al.*: OctaVis: A Virtual Reality System for Clinical Studies and Rehabilitation. In: *Eurographics 2013 – Dirk Bartz Prize, Girone, Spain*, pp. 9–12, 2013. doi: 10.2312/conf/EG2013/med/009-012.

Affiliations

Dariusz Mikolajewski
Kazimierz Wielki University in Bydgoszcz, Institute of Computer Science, Kopernika 1, 85-074 Bydgoszcz, Poland,
Medical University in Lublin, Neuropsychological Research Unit, 2nd Clinic of the Psychiatry and Psychiatric Rehabilitation, Gluska 1, 20-439 Lublin, Poland, darek.mikolajewski@wp.pl

Anna Bryniarska
Opole University of Technology, Faculty of Electrical Engineering, Automatic Control and Informatics, Proszkowska 76, 45-758 Opole, Poland, a.bryniarska@po.edu.pl

Piotr Michal Wilczek
The President Stanislaw Wojciechowski Calisia University, Faculty of Health Sciences, Nowy Swiat 4, 62-800 Kalisz, Poland,
Prof. Zbigniew Religa Foundation for Cardiac Surgery Development, Wolnosci 345a, 41-800, Zabrze, Poland,
Medical Algorithms Sp. z o.o., Legionów 4, 41-902 Bytom, Poland,
p.wilczek@medicalalgorithms.eu

Maria Myslicka
Wroclaw Medical University, Faculty of Medicine, Mikulicza-Radeckiego 5, 50-345 Wroclaw, Poland, mariamyslicka38@gmail.com

Adam Sudol
University of Opole, Faculty of Natural Sciences and Technology, Kardynała Kominka 6/6a, 45-032 Opole, Poland, dasiek@dasiek.info

Dominik Tenczynski
University of Opole, Institute of Medical Sciences, Oleska 48, 45-052 Opole, Poland, esten2000@gmail.com

Michal Kostro
University of Opole, Institute of Medical Sciences, Oleska 48, 45-052 Opole, Poland, michalkostro01@gmail.com

Dominika Rekawek
University of Opole, Institute of Medical Sciences, Oleska 48, 45-052 Opole, Poland, dominikarr01@gmail.com

Rafal Tichy
University of Opole, Institute of Medical Sciences, Oleska 48, 45-052 Opole, Poland, noweczechlo@gmail.com

Rafal Gasz
Opole University of Technology, Faculty of Electrical Engineering, Automatic Control and Informatics, Proszkowska 76, 45-758 Opole, Poland, r.gasz@po.edu.pl

Mariusz Pelc
University of Opole, Institute of Computer Science, Oleska 48, 45-052 Opole, Poland, University of Greenwich, School of Computing and Mathematical Sciences, Old Royal Naval College, Park Row, SE10 9LS London, UK, m.pelc@gre.ac.uk

Jaroslaw Zygarlicki
Opole University of Technology, Faculty of Electrical Engineering, Automatic Control and Informatics, Proszkowska 76, 45-758 Opole, Poland, j.zygarlicki@po.edu.pl

Michal Koziol
Opole University of Technology, Faculty of Electrical Engineering, Automatic Control and Informatics, Proszkowska 76, 45-758 Opole, Poland, m.koziol@po.edu.pl

Radek Martinek
VSB-Technical University Ostrava, Department of Cybernetics and Biomedical Engineering – FEECS, 17. listopadu 2172/15, 708 00 Ostrava–Poruba, Czech Republic,
Opole University of Technology, Faculty of Electrical Engineering, Automatic Control and Informatics, Proszkowska 76, 45-758 Opole, Poland, radek.martinek@vsb.cz

Radana Kahankova Vilimkova
VSB-Technical University Ostrava, Department of Cybernetics and Biomedical Engineering – FEECS, 17. listopadu 2172/15, 708 00 Ostrava–Poruba, Czech Republic,
Opole University of Technology, Faculty of Electrical Engineering, Automatic Control and Informatics, Proszkowska 76, 45-758 Opole, Poland, radana.vilimkova.kahankova@vsb.cz

Dominik Vilimek
VSB-Technical University Ostrava, Department of Cybernetics and Biomedical Engineering – FEECS, 17. listopadu 2172/15, 708 00 Ostrava–Poruba, Czech Republic, dominik.vilimek@vsb.cz

Aleksandra Kawala-Sterniuk
Opole University of Technology, Faculty of Electrical Engineering, Automatic Control and Informatics, Proszkowska 76, 45-758 Opole, Poland, kawala84@gmail.com

Received: 21.07.2023
Revised: 29.01.2024
Accepted: 04.02.2024

Computer Science • 25(1) 2024 https://doi.org/10.7494/csci.2024.25.1.5690

Miłosz Zdybał
Marcin Kucharczyk
Marcin Wolter

MACHINE LEARNING BASED EVENT RECONSTRUCTION FOR THE MUonE EXPERIMENT

Abstract *A proof-of-concept solution based on the machine learning techniques has been implemented and tested within the MUonE experiment designed to search for New Physics in the sector of anomalous magnetic moment of a muon. The results of the DNN based algorithm are comparable to the classical reconstruction, reducing enormously the execution time for the pattern recognition phase. The present implementation meets the conditions of classical reconstruction, providing an advantageous basis for further studies.*

Keywords machine learning, artificial neural networks, track reconstruction, high energy physics

Citation Computer Science 25(1) 2024: 147–168

1. Introduction

Significant developments have been applied during the last decades in the field of High Energy Physics (HEP) experiments, including computing technologies. Searches for New Physics phenomena, being an expansion of the so-called Standard Model, i.e. current incomplete theoretical knowledge about the basic behavior of the fundamental constituents of nature and the interactions between them, lead to experimental studies carried out at ever increasing energies. The number of particles created by the interaction of two particles (collision event) is generally increasing with the collision energy. As a consequence a huge number of charged particles have to be reconstructed (e.g. in proton-proton collisions), resulting in much more complex event patterns. A typical event in proton-proton collision showing the tracks of multiple particles passing through the detector is presented in Figure 1, where the particles leave energy deposits (hits) in consecutive detector layers, being a basis for further track reconstruction. In order to enable the search for rare interesting collision events immersed in a huge background of events exhibiting well-known physics, the data rates related to the detector luminosity[1] have increased enormously (e.g., 40 MHz readout rate in LHC). It has to be reduced online by more than five orders of magnitude before the information from an event is written on mass storage for further analysis.

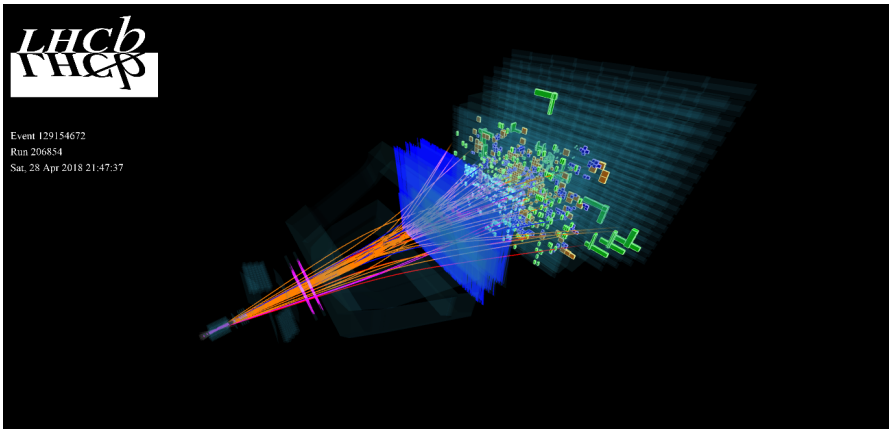


Figure 1. Example of an event in High Energy Physics experiment, showing tracks of multiple particles passing through the detector [31]

This paper aims to review the machine learning based approach applied in crucial stages of the data analysis process in HEP experiments, i.e. the procedure to determine basic kinematic parameters of charged particles at their point of production and the procedure to establish the location of these production points. They are

[1]Luminosity translates to the number of collisions per second and it is related to track density.

commonly called track and vertex reconstruction. High density of tracks in a single collision event (detector occupancy) in operating and planned high-energy physics experiments results in a large combinatorics of hits in the event pattern recognition. Therefore, a novel machine learning based event reconstruction algorithms have been developed and tested within a framework of the MUonE experiment [2] in order to maximize the statistical power of the final physics measurement. The results of the DNN based algorithm are comparable to the classical reconstruction, allowing not only to reduce execution time of the pattern recognition phase, but also to improve the precision and efficiency of the track and vertex reconstruction.

2. Particle track reconstruction in High Energy Physics experiments

2.1. State of the art

In High Energy Physics experiments the reconstruction of charged particle trajectories in the detectors is the most crucial process as it constitutes the major part of reconstruction time of the whole event. Tracking algorithms partition a collection of position measurements into groups corresponding to the to the hits originating from the same particle traversing through the detector. In the next step the parametrized trajectories are fitted to these collections to extract particle kinematics and locations of interaction vertices. The obtained results are later combined with measurements from other detector systems, like calorimeters measuring the particle energy, to construct a complete physical model of an event. In the last stage of the data analysis the reconstructed events are used to extract the physical quantities.

Traditional tracking algorithms have been used with great success in the HEP experiments. However they suffer from serious limitations that motivate for searching new solutions. These algorithms are inherently serial, and scale poorly with detector occupancy. In particle physics the measurement of charged particle parameters is one of the most computationally-intensive processes. This process relies on measurements of particle tracking detectors to construct a particle trajectory by combining the detected hits and resolving the particle momentum via fitting the trajectory points using the Kalman filter [24], The Kalman filter processes a set of discrete measurements to determine the internal state of a linear dynamical system (see Fig. 2). Both the measurements and the system can be subjected to independent random perturbations or noise. By combining predictions based on the previous state estimates with subsequent measurements, the impact of these perturbations on the following state estimates can be minimized.

In high luminosity experiments, where multiple particles are produced as a result of an interaction, the process of isolating detector hits for each particle trajectory relies on considering each combination of hits that can potentially form a track and then fitting each hypothesis to determine which one represents a valid trajectory. Also the noise, always present in particle tracking detectors and resulting in additional hits

in the detectors, increases the number of possible combinations. This process can be time-consuming, amounting to the most significant part of the total data post-processing time. For such methods based on the Kalman filter the CPU needed for track reconstruction grows rapidly with the luminosity (see Fig. 3). Therefore, more advanced methods of finding particle trajectories using the measurements from all active detector elements can be investigated.

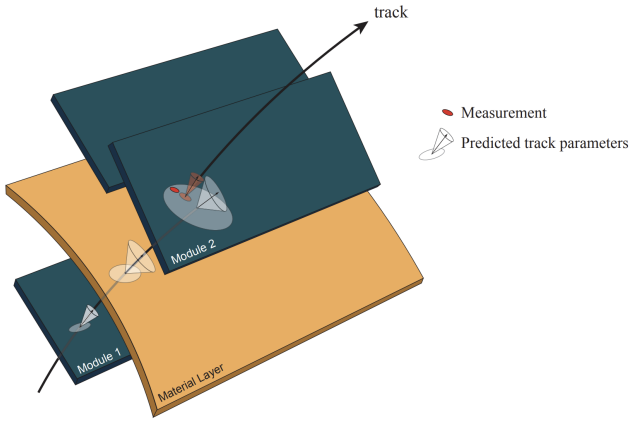


Figure 2. Simplified illustration of a typical extrapolation process within a Kalman filter. The track representation on the detector module 1 is propagated onto the next measurement surface, which results in the track prediction on module 2 [37]

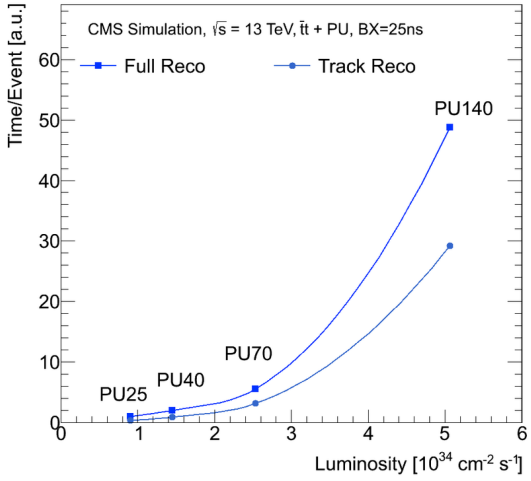


Figure 3. Expected CPU time per event as a function of instantaneous luminosity collected by the CMS experiment, for both full reconstruction and the dominant tracking part. The pile-up (PU) is the number of interactions per beam crossing. PU25 corresponds to the data taken in 2012, and PU140 corresponds to the HL-LHC era. The CPU time of the reconstruction is dominated by the track reconstruction [12]

2.2. Machine learning based track reconstruction

Machine learning methods such as deep neural networks have some promising characteristics that could prove effective for particle tracking. Neural networks are known to be very good at finding patterns and modeling non-linear dependencies in data. They also involve highly regular computation that can run effectively on parallel architectures such as GPUs (Graphics Processing Units).

Neural Networks are therefore widely used in High Energy Physics not only as a classification tool, but also for other tasks like intelligent data reduction and time series analysis [27] or reconstruction of a pulse shape from the front-end electronics [26]. They are used as well to optimize processes in various environments (Reinforcement Learning), for example automate the management of resources in a computing cloud [20].

The first approach to use neural networks for particle track reconstruction was done already in the 1980s [34]. However modern techniques based on deep learning have started to be studied in the last years. Two categories of machine learning solutions, image-based and point-based models were investigated [17].

The Convolutional Neural Networks (CNNs) (see for example [13] for description of various types of neural networks) proved to be a very efficient tool in image recognition. As such they are widely used in physics research, one of the examples is the CREDO experiment offline trigger using CNN to tag artefacts appearing in the CREDO database [35].

The computer vision techniques based on CNN such as semantic segmentation and image captioning have inspired the image-based models of particle track recognition. In this approach the detector data is treated as an image and the convolutional and also recurrent neural networks are applied to detect tracks.

Image based track reconstruction

In the image-based approach multiple track finding problem might be treated in a similar fashion to the image captioning, where the descriptions of the tracks (i.e., track parameters) are analogous to the text captions assigned to the various patterns seen in the image [39]. For this purpose, the long/short-term memory (LSTM) [21] layer is used.

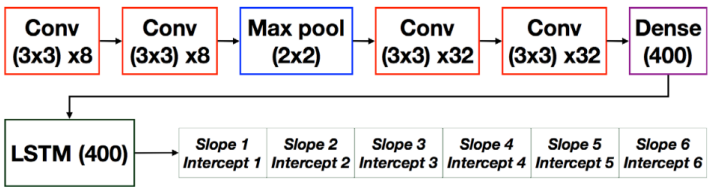


Figure 4. Convolutional deep neural network with convolutional layers followed by dense and LSTM layers. Network is trained to reconstruct track parameters for multiple track events [29]

In the case of track reconstruction, the Convolutional Neural Network (CNN) sequentially reconstructs consecutive tracks, which are the sequential input for the LSTM layer. Example of such a network suited to reconstruct events with multiple tracks is shown in Figure 4.

The image-based models map nicely onto well-studied problems in computer vision and sequence modeling. However, when scaling up to the realistic complexity of particle physics experiments they are suffering from high dimensionality and sparsity.

Point based track reconstruction

In the point-based models, the continuously distributed spacepoint hits are used. They are structured in a list or tree for learning how to group them into track candidates. A recurrent neural network acts as an iterative filter similar to a Kalman Filter. Therefore the model predicts the position of the point on the next detector layer. It can be used to build tracks by selecting the closest spacepoint to the prediction at every layer and searches until a complete track is found. This architecture might use an LSTM layer and fully connected layer with linear activation function to reconstruct multiple tracks.

Graph Neural Network models

In 2020 the Exa.TrkX project [22] has demonstrated the applicability of Geometric Deep Learning (GDL) methods to particle tracking [14] (specifically Graph Neural Networks (GNNs) [38]). GNNs have already proven to be succesfull in computer vision applications [28]. Such a network is concerned with learning representations of data that have complex geometrical relationships and no natural ordering, which corresponds well with hits in the detector. In addition such models are naturally parallel and therefore well-suited to run on hardware accelerators and GPUs. The training of such a network might be computational demanding, but an answer of a trained network is fast and the computer time needed increases linearly with the number of tracks.

In applications to track finding graphs are constructed from the cloud of hits in each event. Edges are drawn between hits that may come from the same particle track according to some loose heuristic criteria. The GNN model is then trained to classify the graph edges as real or fake, giving a pure and efficient sample of track segments which can be used to construct full track candidates. Advanced studies concerning the application of GNNs for track reconstructions were presented by both CMS [9] and ATLAS [11] experiments.

It was shown [23], that within the simplifying assumptions, the GNN based track finding algorithm can meet the tracking performance requirements of current, high luminosity collider experiments. This performance should be robust against systematic effects like detector noise, misalignment, and pile-up. The GNN based algorithms are promising and growing in popularity.

3. MUonE experiment

A very promising opportunity to search for New Physics in the sector of muon's anomalous magnetic moment a_μ has appeared with the new results from $g-2$ experiments [6, 8], which measured the anomaly with respect to the Standard Model prediction at the level of 4.2 standard deviations (see Fig. 5).

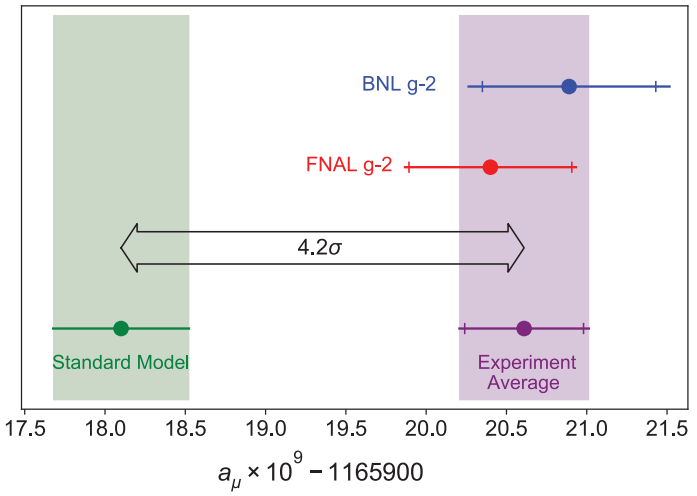


Figure 5. Comparison of the measurements of anomalous muon magnetic moment a_μ with the Standard Model prediction [6], where the discrepancy of 4.2 standard deviation between theory and experiment can be observed. In order to improve the visibility of the discrepancy the unit on the x-axis corresponds to a given a_μ value multiplied by 10^9 and subtracted with 1165900

As the main limitation of eventual discovery comes from the precision of the theoretical Standard Model predictions, dominated by the uncertainty related to the hadronic interactions, the idea is to use the process of elastic muon scattering on electrons for the precise estimation of the hadronic contribution to a_μ. The experiment dedicated to measure precisely such a hadronic contribution is the MUonE project [2], designed to determine the hadronic part of the running of the electromagnetic coupling constant in the space-like region by the scattering of high-energy muons on atomic electrons in a low-Z target through the elastic process $\mu e \to \mu e$ [4]. A result with significantly suppressed statistical uncertainty can be achieved on the hadronic contribution to a_μ, which, together with the results from running [6] or planned [5] $g-2$ experiments supposed to measure directly a_μ, would increase the significance of observed discrepancy up to the level of 7 standard deviations. In order to measure the hadronic contribution with a required accuracy, a significant boost in precision and event statistics is necessary. This can only be achieved by accurate performance of the both the trigger and tracking system of the MUonE experiment.

3.1. Experimental setup

The data samples of $\mu e \to \mu e$ elastic scattering will be collected in MUonE experiment using 150–160 GeV muons impinging on the atomic electrons of Beryllium targets. The upgraded M2 muon beam at the CERN SPS [15] will be used for this purpose, delivering high energy and high-intensity muon and hadron beams, and also low intensity electron beams for calibration. The beams are conducted in the following way. First, the SPS primary proton beam of 450 GeV impinges on a primary Beryllium production target, where mainly secondary protons, electrons, pions and kaons are produced. In the next step the secondary particles are transported in a beam line allowing the pions and kaons to decay into muons. At this stage a 9.9 m thick Beryllium absorber stops the left-over hadrons, allowing the muons at the same time to pass basically unharmed. Next, such muons are momentum-selected employing large magnetic collimators, and finally the muons with momenta in the range of 100 and 225 GeV/c are selected. The typical maximal intensity for a beam energy of 160 GeV is $5 \cdot 10^7$ μ/sec.

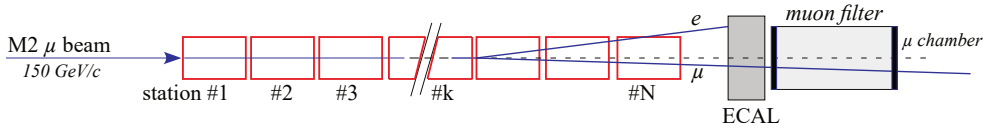


Figure 6. Schematic view of the MUonE experimental apparatus [2]

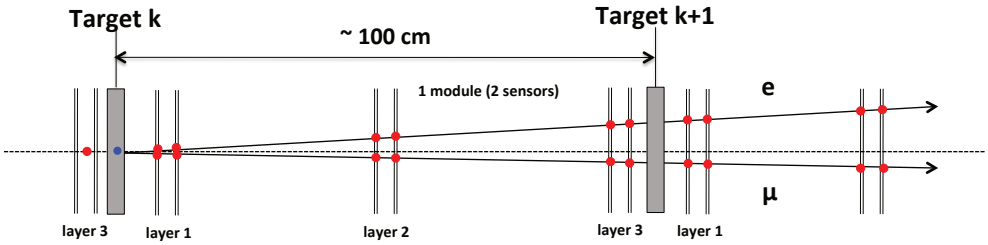


Figure 7. Schematic view of a single tracking station [2]

The main detectors of the MUonE experiment [2] are specified in Figure 6 and 7. The tracking system will provide the precise measurement of the scattering angles of the outgoing electron and muon, with respect to the direction of the incoming muon beam. It will contain 40 identical stations (see Fig. 6), each consisting of a 3 cm thick layer of Beryllium coupled to 3 Si layers (see Fig. 7) located at a relative distance of about one meter from each other and spaced by intermediate air gaps. Such an arrangement provides both a distributed target with low-Z and the tracking system. The silicon strip sensors for the MUonE project are characterized by a large active area sufficient to cover the full MUonE required acceptance, together with appropriate

spacial resolution. They can also support the high readout rate of 40 MHz required for MUonE with their accompanying front-end electronics. The downstream particle identifiers are planned to be installed, required to solve the muon-electron ambiguity. That will be a calorimeter for the electrons and a muon filter for the muons. A homogeneous electromagnetic calorimeter placed downstream all the tracker stations will be used, in order to accomplish the physical requirements, i.e. particle identification, measurement of the electron energy and event selection.

3.2. MUonE test beam in 2018

In 2018 the MUonE test run was performed with the aim to provide information for the design of the final MUonE detector setup [3]. The apparatus used was situated in the EHN2 experimental area, located behind the COMPASS spectrometer. The 187 GeV positive muon beam was obtained from decays of pions, which were stopped in the beam dump in the posterior part of the spectrometer. A 10×10 cm^2, 8 mm thick graphite target was followed by the tracking system with 16 microstrip layers consisted of a $9.293 \times 9.293 \times 0.041$ cm^3 single-side sensor with 384 channels (see Fig. 8). Each tracking layer measured one hit coordinate, x or y. Despite that, stereo stations rotated by an angle $\pm\pi/4$ were added. A calorimeter located at the end of the system, composed from BGO tapered crystals, covered an angular acceptance of about 15 mrad on each side from the center of Si layers.

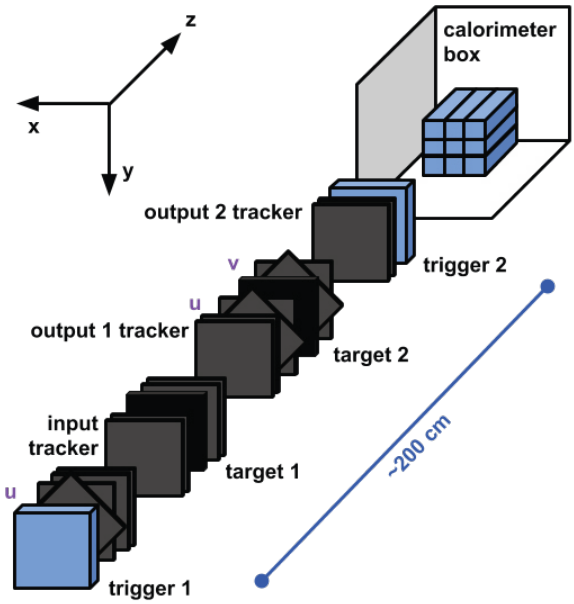


Figure 8. Schematic view of the apparatus used in the MUonE 2018 test run [3]

From the data collected at the last period of the 6 months run, after the final requirements on the presence of an incoming track and at least two outgoing tracks, the number of events used in the analysis was reduced to $\sim 94 \cdot 10^3$. The event reconstruction consisted of the following main steps: the pattern recognition, 2-dimensional track finding, combining 2-dimensional track candidates into a 3-dimensional track and finally, constructing a scattering event from three 3-dimensional tracks with a dedicated kinematic vertex fit based on a constrained least square method. Finally, this allowed to obtain a clean sample of μe elastic scattering events.

The angular resolution was determined with the simulation, which met the MUonE 2018 test beam configuration. A sample of $\sim 100 \cdot 10^3$ events was produced, where the incoming muon beam was assumed to be a monoenergetic beam with energy of 187 GeV, and x and y distributions were adjusted to match the ones measured with data.

4. Machine learning based track reconstruction for MUonE

The measurement of the hadronic contribution to a_μ in the MUonE experiment requires novel fast and efficient real-time based algorithms for the track and vertex reconstruction, together with a flexible trigger system. New event reconstruction methods developed for the MUonE experiment, based on the novel hardware-triggerless techniques or, alternatively, on machine learning methods implemented on parallel GPU processing, may become a standard approach in the future High Energy Physics experiments, facing an enormously tight execution time imposed by a fully software trigger system and achieving a maximum possible event reconstruction efficiency and precision. It will allow to efficiently reduce the size of data expected to increase fastly in the future experiments, and also to maximize the statistical power of the final physics measurement. In the MUonE experiment the algorithms of track finding based on machine learning techniques are being developed and tested in order to speed up the reconstruction process. This may lead to the significant acceleration of the execution of pattern recognition algorithms in the real-time event reconstruction algorithms. Moreover, the use of DNN techniques may significantly improve both the reconstruction efficiency and the precision of measuring the parameters that are crucial for final measurement.

4.1. Two-dimensional machine learning based reconstruction

A first approach to use the machine learning techniques in context of the MUonE experiment [29] was based on an image-based model and a convolutional neural network [30]. This type of neural network is predominantly used in computer vision tasks, using a set of filters (called kernels) that analyze the image with a relatively small perception window (few pixels in size) that scans the input to produce the activation map of the filter. Network creates a set of filters sensible to different features in the input. Single filter can find multiple instances of the feature in the input image.

Training and testing datasets were generated using a two-dimensional toy-model. In total $40 \cdot 10^3$ events corresponding to an elastic scattering signal with the MUonE 2018 test beam configuration (see Sec. 3.2) were produced. Each event contained one or two tracks reconstructed with the linear fit and was represented with a two-dimensional 28×28 pixel image. Optionally, the noise was also included.

The neural network was implemented in KERAS [25] with TensorFlow [1] backend. It was trained to respond to the input image with slopes and intercepts of the tracks. Two convolutional layers were used with 3×3 convolution window, followed by the MaxPooling layer and another two convolutional layers. The dropout layer was used to control the overtraining. Final regression was performed using the 1024-node dense layer. The full network had over 2 million trainable parameters. For multi-track events, long/short-term memory (LSTM) mechanism was used. It was inspired by the work of HEP.TrkX project [16, 18]. To make events more realistic, noise and pixel inefficiency was introduced. Noise could be defined in the 0–30% range, meaning a probability of pixel not belonging to the track to generate a signal. Pixel efficiency was lowered to 70% by changing the probability of track pixel to generate a signal.

Results provided by the CNN were used to find hits closest to the track candidates. The tracks were reconstructed using linear robust fit [10]. Differences between reconstructed and true tracks are shown in Figure 9, including 10% and 30% noise levels. Results were compared with the classical reconstruction algorithm (included in the plots). The neural network based approach proved to be successful and prompted the further development using three-dimensional approach.

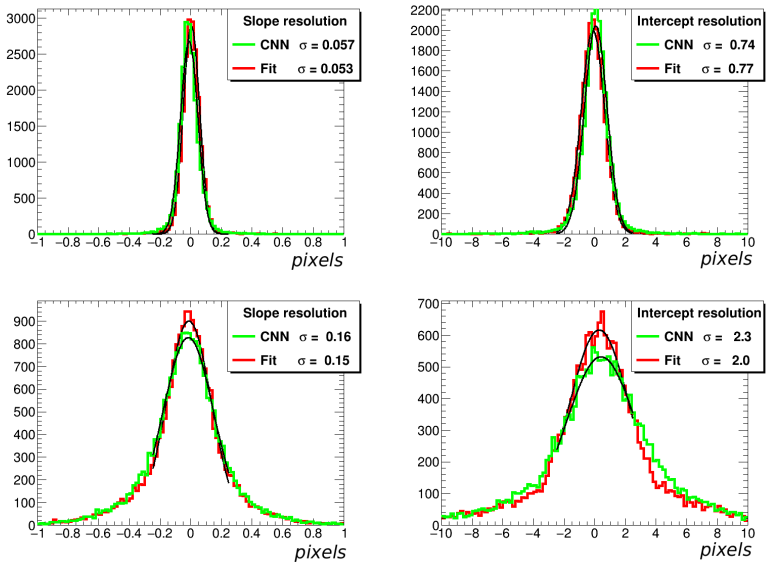


Figure 9. Comparison of the distributions of the difference between the reconstructed and true track parameters: slope (left) and intercept (right), between CNN-based and classical track reconstruction. Noise level at 10% (top) and 30% (bottom). Figure taken from [29]

4.2. Three-dimensional DNN based track reconstruction

A natural next step after the work described in the previous section was an implementation of the machine learning based approach into three dimensions [40], that would meet much better the requirements of the MUonE experiment. In general an artificial neural network was designed to reproduce properly the track parameters. Based on the set of hit coordinates, the network's task is to predict the slope and the intercept of the track for each of two outgoing μ-e elastic scattering signal particles.

4.2.1. Learning dataset

The tracking detector of the MUonE experiment (in the final detector configuration as well as in the 2018 test run) is based on silicon strip sensors which provide only one value for the measurement that can be combined with the sensor position along the beam axis to create two-dimensional representation of the hit position, i.e. (x, z) or (y, z). An assumption was imposed that input data for the network will not include the information about the type of the hit (x, y, or u, v for stereo layers), but all the hits will be ordered by an increasing z coordinate.

The training dataset was generated using a leading-order event generator with the detector simulation preformed with GEANT4 [7]. The sample contained about $100 \cdot 10^3$ events corresponding to the MUonE 2018 test beam setup described in Sec. 3.2. Input vector of the neural network consisted of 20 floating point values representing the measurements of the detector. Hits were arranged by the increasing z coordinate, without distinction between x, y and *stereo* hits. The z coordinates were not included in the input vector, as they were identical in all events. For each event a ground truth was provided in the form of slope and intercept of outgoing tracks, 12 values in total. The dataset was split into training and testing subsets in the $4:1$ ratio.

4.2.2. Artificial neural network

The PyTorch [32] was chosen as the machine learning framework, as it incorporates tools needed for data handling, training process and inference, all with GPU support to accelerate underlying matrix-based computations. The input vector contained collection of hits described in the previous section. To reduce its size as well as the size of correlated network and its complicity, information not critical for the track reconstruction was removed. Hits were sorted by ascending z value, which made explicit use of this coordinate, repeating in all events, redundant. In addition, hits related to the incoming muon were skipped, as the algorithm focused on the reconstruction of outgoing tracks. Final input vector included 20 values, each representing a measurement made by a silicon strip sensor. There was no distinction among x, y and *stereo* hits. The output vector contained slopes and intercepts of the two outgoing tracks, represented in x-z and y-z projections, totalling in 8 values. For ease of comparison, this is the same format as the ground truth was provided in.

The artificial neural network consisted of four fully connected layers of 1000 neurons each, with additional layers for input and output (see Fig. 10). It is important to mention that at this stage of development no hyperparameter optimization was performed, i.e. parameters of the network were arbitrarily set to achieve acceptable results in reasonable time during the development. The *MSELoss* (Mean Squared Error Loss) [36] was chosen as the loss function, its implementation from the PyTorch package was used. In the process of training, the network was optimized in a way that minimized the mean squared error between the output and ground truth. In terms of the activation function, ReLU (rectified linear unit) was used, as being suitable for deep neural networks.

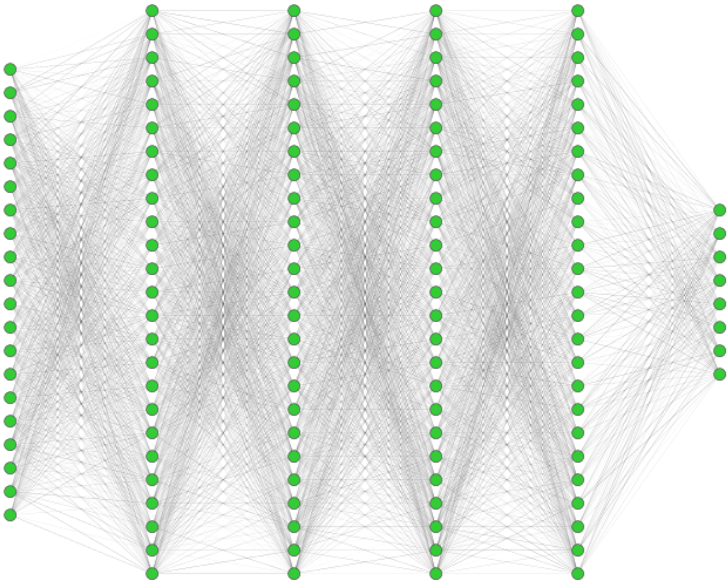


Figure 10. Architecture of the neural network used for track reconstruction. The input layer has 20 nodes, the following four hidden layers have 1000 nodes each and the output layer consists of 8 nodes. Number of nodes in the hidden layers in plot was reduced from 1000 to 25 for clarity

4.2.3. Reconstruction algorithm

Comparison of the DNN-predicted tracks with the ground truth revealed that tracks are relatively close to each other (see Fig. 11), however reconstruction did not provide a precision required in the experiment. More complex algorithm had to be developed, where DNN was responsible only for the pattern recognition, being the most CPU time-consuming stage of the reconstruction in comparison to relatively fast linear fitting. Proposed algorithm is presented in Figure 12 and described in the following sections.

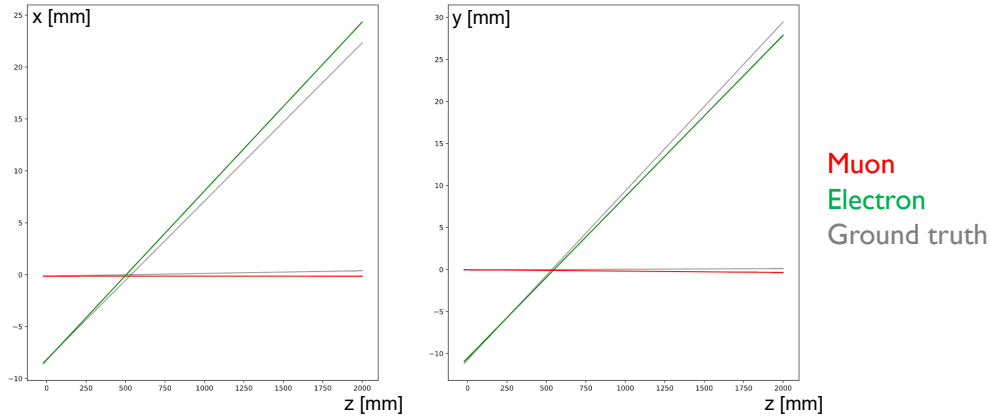


Figure 11. Example of the tracks reconstructed by the DNN, before applying further steps of the reconstruction algorithm. The ground truth shown in grey

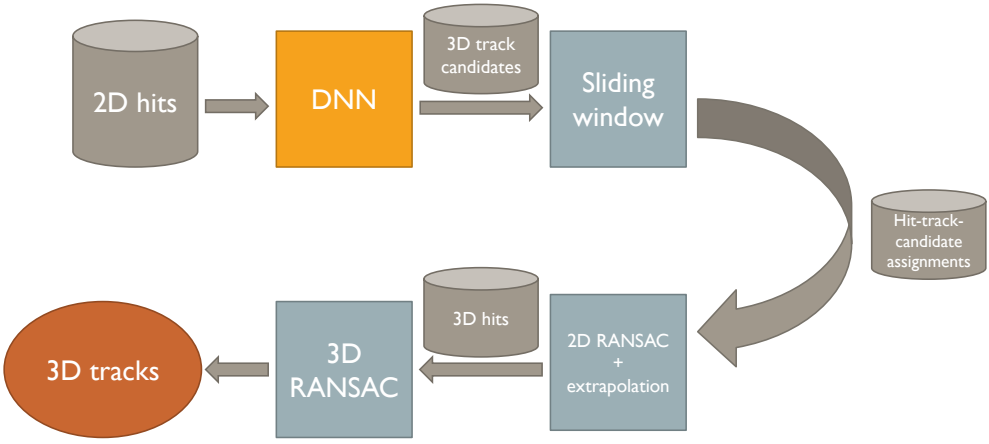


Figure 12. The DNN-based algorithm for track reconstruction

- Deep neural network based pattern recognition.
  In the first step, the DNN machine learning model was used to turn the collection of hits representing μ-e elastic scattering signal event into three-dimensional track candidates. Every hit was then assigned to the DNN-reconstructed track that was the closest geometrically in its plane. Example of the event with hits assigned to the tracks is shown in Figure 13.

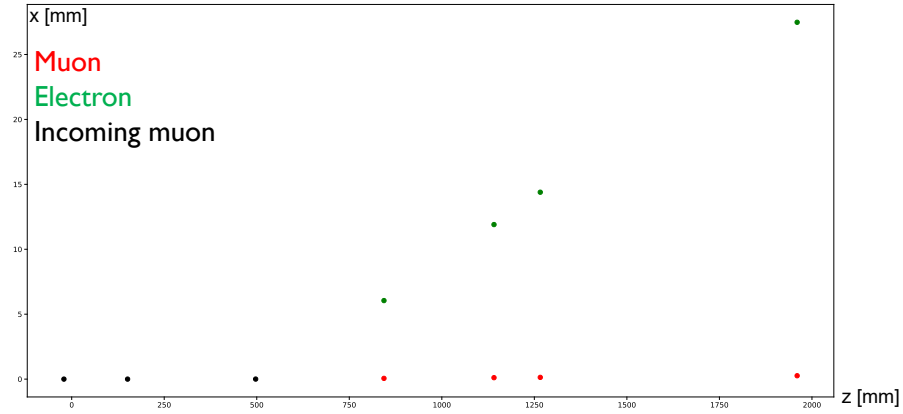


Figure 13. Example of the collections of hits corresponding to the μ-e elastic scattering signal tracks constructed based on the DNN-predicted track candidates. Points represent the hits in x-z projection, colours correspond to the particle type

- Two-dimensional linear fit.
  At this stage, RANSAC method (*Random Sample Consensus*) [19] was used to reconstruct two-dimensional temporary tracks in z-x and z-y projections. The RANSAC iterative algorithm was chosen as it is effective for outlier removal. Implementation provided in Scikit-learn package [33] was used. As a result, two 2D lines were established in both z-x and z-y planes. Example of temporary tracks resulting from the 2D linear fit in z-x projection is shown in Figure 14.

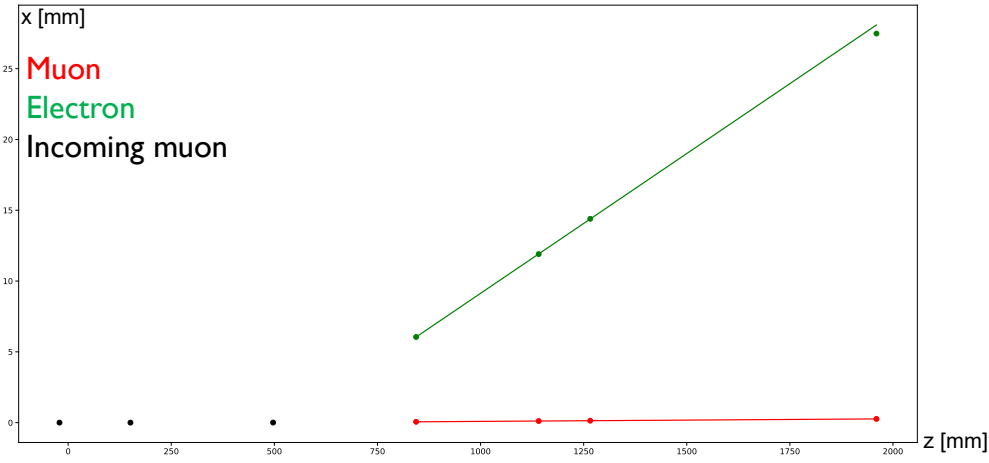


Figure 14. Example of the result of the linear fit (solid lines) in x-z projection for an outgoing muon and electron from the μ-e elastic scattering signal event. The points represent the hits in x-z projection, colours correspond to the particle type

- Final 3-dimensional track fit.
  Two-dimensional temporary tracks from the previous step were used to extrapolate the missing coordinate for each hit. With a collection of 3D hits assigned to every track, the final linear fit was performed with 3-dimensional RANSAC algorithm. Example of reconstructed three-dimensional tracks is shown in Figure 15.

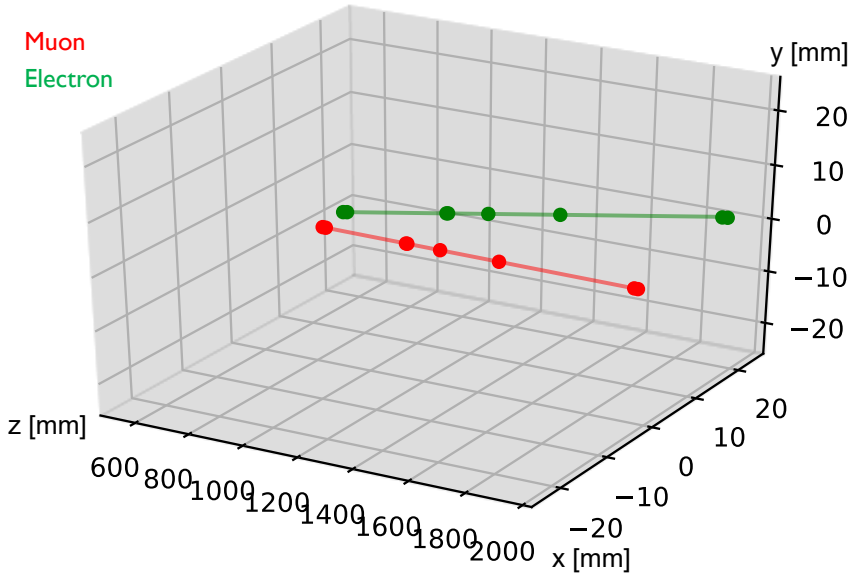


Figure 15. Example of the result of the 3D linear track fit (solid lines – red for muon and green for electron) for an outgoing muon and electron from the μ-e elastic scattering signal event. The points represent the hits, colours correspond to the particle type (red for muon and green for electron)

4.2.4. Results

To assess the quality of the reconstructed tracks, the resolutions in track slopes were determined. To achieve this, histograms of difference in slopes between reconstructed track and ground truth were fitted with the Gaussian distribution, and the value of the standard deviation was interpreted as the resolution. In the case of electron track double Gaussian was used as this particle is more affected by the multiple scattering. The same procedure was performed for the tracks reconstructed with classical algorithm. Distributions of the slope differences are shown in Figure 16 and resolutions are summarized in Table 1. Additionally, efficiencies of the reconstruction algorithms were compared. Efficiency in this case is defined as the percentage of the tracks with the slope calculated with difference less than $1 \cdot 10^{-2}$ when compared to the ground truth. Efficiencies of reconstruction algorithms are compared in Table 2. Results achieved by the DNN-based algorithm are on pair with the conventional algorithm. Differences, if

present, are not significant. This shows that the machine learning approach to track reconstruction has a great potential and should be investigated further.

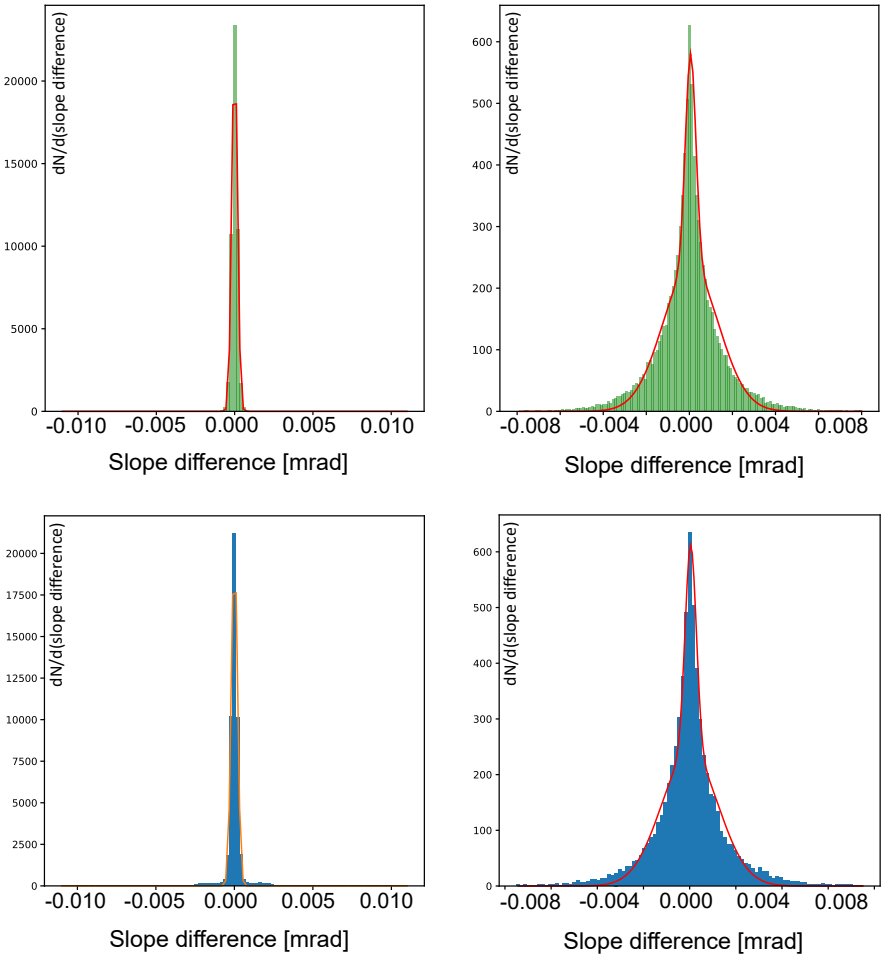


Figure 16. Distributions of slope difference of reconstructed tracks (left for muons, right for electrons) in relation to the MC truth, for DNN-based algorithm (upper plots) and classical reconstruction (bottom plots)

Table 1
Slope resolutions for an outgoing muon and electron

Particle	DNN based [mrad]	Classical [mrad]
Muon	$\sigma = 0.000018$	$\sigma = 0.000019$
Electron	$\sigma_1 = 1.290$	$\sigma_1 = 1.230$
	$\sigma_2 = 0.245$	$\sigma_2 = 0.244$

Table 2
Efficiencies of reconstruction algorithms, as defined in the text

Particle	DNN based [%]	Classical [%]
Muon	100	99.98
Electron	99.66	99.38

5. Summary and outlook

The present DNN based algorithm prototype for the three-dimensional track reconstruction in the MUonE experiment proved to be competitive with the classical track reconstruction tasks in terms of quality, with potential performance benefits. Further development will involve the implementation of the neural network architecture based on Graph Neural Network. Graph Neural Networks (GNNs, subsection 2.2) ensure, among others, the inductive bias, reduction of number of parameters, more elaborated loss function, and above all a much more natural data representation. New event reconstruction methods being developed for the MUonE experiment, based on the novel hardware-triggerless techniques using machine learning methods may become a standard approach in the future High Energy Physics experiments, facing an enormously tight execution time imposed by a fully real-time reconstruction and achieving a maximum possible efficiency and precision. Although such techniques are not yet applied on a scale in any High Energy Physics experiments, they are intensively developed and planned to be employed in the near future.

Acknowledgements

This research was supported by the National Science Centre NCN (Poland) under the contract no. 2022/45/B/ST2/00318 and also supported in part by PL-Grid Infrastructure.

References

[1] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., *et al.*: TensorFlow: Large-scale machine learning on heterogeneous distributed systems, *arXiv preprint arXiv:160304467*, 2016.

[2] Abbiendi G.: Letter of Intent: the MUonE project, *CERN-SPSC-2019-026, SPSC-I-252*, 2019. https://cds.cern.ch/record/2677471.

[3] Abbiendi G., Ballerini G., Banerjee D., Bernhard J., Bonanomi M., Brizzolari C., Foggetta L., *et al.*: A study of muon-electron elastic scattering in a test beam, *Journal of Instrumentation*, vol. 16(06), P06005, 2021. doi: 10.1088/1748-0221/16/06/p06005.

[4] Abbiendi G., Carloni Calame C.M., Marconi U., Matteuzzi C., Montagna G., Nicrosini O., Passera M., *et al.*: Measuring the leading hadronic contribution to the muon g-2 via μe scattering, *The European Physical Journal C*, vol. 77(3), 2017. doi: 10.1140/epjc/s10052-017-4633-z.

[5] Abe M., Bae S., Beer G., Bunce G., Choi H., Choi S., Chung M., *et al.*: A new approach for measuring the muon anomalous magnetic moment and electric dipole moment, *Progress of Theoretical and Experimental Physics*, vol. 2019(5), 053C02, 2019. doi: 10.1093/ptep/ptz030.

[6] Abi B., Albahri T., Al-Kilani S., Allspach D., Alonzi L.P., Anastasi A., Anisenkov A., *et al.*: Measurement of the Positive Muon Anomalous Magnetic Moment to 0.46 ppm, *Physical Review Letter*, vol. 126, 141801, 2021. doi: 10.1103/PhysRevLett.126.141801.

[7] Agostinelli S., Allison J., Amako K., Apostolakis J., Araujo H., Arce P., Asai M., *et al.*: GEANT4–a simulation toolkit, *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506(3), pp. 250–303, 2003. doi: 10.1016/S0168-9002(03)01368-8.

[8] Bennett G.W., Bousquet B., Brown H.N., Bunce G., Carey R.M., Cushman P., Danby G.T., *et al.*: Final report of the E821 muon anomalous magnetic moment measurement at BNL, *Physical Review D*, vol. 73(7), 072003, 2006. doi: 10.1103/PhysRevD.73.072003.

[9] Bhattacharya S., Chernyavskaya N., Ghosh S., Gray L., Kieseler J., Klijnsma T., Long K., *et al.*: GNN-based end-to-end reconstruction in the CMS Phase 2 High-Granularity Calorimeter, 2022. doi: 10.48550/ARXIV.2203.01189.

[10] Brun R., Rademakers F.: ROOT – An object oriented data analysis framework, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 389(1), pp. 81–86, 1997. doi: 10.1016/S0168-9002(97)00048-X.

[11] Caillou S., Calafiura P., Farrell S.A., Ju X., Murnane D.T., Rougier C., Stark J., Vallier A.: ATLAS ITk Track Reconstruction with a GNN-based pipeline. Technical report: ATL-COM-ITK-2022-057, 2022. https://cds.cern.ch/record/2815578/.

[12] Cerati G., Elmer P., Krutelyov S., Lantz S., Lefebvre M., McDermott K., Riley D., *et al.*: Kalman filter tracking on parallel architectures, *Journal of Physics: Conference Series*, vol. 898, 042051, 2017. doi: 10.1088/1742-6596/898/4/042051.

[13] Chollet F.: *Deep learning with Python*, Simon and Schuster, 2021.

[14] Choma N., Murnane D., Ju X., Calafiura P., Conlon S., Farrell S., Cerati G., *et al.*: Track seeding and labelling with embedded-space graph neural networks, *arXiv preprint arXiv:200700149*, 2020.

[15] Doble N., Gatignon L., Holtey von G., Novoskoltsev F.N.: The upgraded muon beam at the SPS, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 343(2–3), pp. 351–362, 1993. doi: 10.1016/0168-9002(94)90212-7.

[16] Farrell S., Anderson D., Calafiura P., Cerati G., Gray L., Kowalkowski J., Mudigonda M., *et al.*: The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking, *EPJ Web of Conferences*, vol. 150, 00003, 2017. doi: 10.1051/epjconf/201715000003.

[17] Farrell S., Calafiura P., Mudigonda M., Mr. Prabhat, Anderson D., Bendavi J., Spiropoulou M., *et al.*: Particle Track Reconstruction with Deep Learning. In: *31st Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. https://dl4physicalsciences.github.io/files/nips_dlps_2017_28.pdf.

[18] Farrell S., Calafiura P., Mudigonda M., Prabhat, Anderson D., Vlimant J.R., Zheng S., *et al.*: Novel deep learning methods for track reconstruction, 2018. doi: 10.48550/ARXIV.1810.06111.

[19] Fischler M.A., Bolles R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Commun ACM*, vol. 24(6), pp. 381–395, 1981. doi: 10.1145/358669.358692.

[20] Funika W., Koperek P., Kitowski J.: Continuous self-adaptation of control policies in automatic cloud management, *Concurrency and Computation: Practice and Experience*, vol. 35(20), e7371, 2023.

[21] Hochreiter S., Schmidhuber J.: Long short-term memory, *Neural Computation*, vol. 9(8), pp. 1735–1780, 1997.

[22] Ju X., Farrell S., Calafiura P., Murnane D., Prabhat, Gray L., Klijnsma T., *et al.*: Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors. In: *33rd Annual Conference on Neural Information Processing Systems*, 2020. doi: 10.48550/arXiv.2003.11603.

[23] Ju X., Murnane D., Calafiura P., Choma N., Conlon S., Farrell S., Xu Y., *et al.*: Performance of a geometric deep learning pipeline for HL-LHC particle tracking, *The European Physical Journal C*, vol. 81, 876, 2021. doi: 10.1140/epjc/s10052-021-09675-8.

[24] Kalman R.E.: A New Approach to Linear Filtering and Prediction Problems, *Journal of Basic Engineering*, vol. 82(1), pp. 35–45, 1960. doi: 10.1115/1.3662552.

[25] Keras, https://keras.io, 2015.

[26] Kopciewicz P., Akiba K.C., Szumlak T., Sitko S., Barter W., Buytaert J., Eklund L., *et al.*: Simulation and optimization studies of the LHCb beetle readout ASIC and machine learning approach for pulse shape reconstruction, *Sensors*, vol. 21(18), 6075, 2021. doi: 10.3390/s21186075.

[27] Kopciewicz P., Szumlak T., Majewski M., Akiba K., Augusto O., Back J., Bobulska D.S., *et al.*: The upgrade I of LHCb VELO – towards an intelligent monitoring platform, *Journal of Instrumentation*, vol. 15(06), C06009, 2020. doi: 10.1088/1748-0221/15/06/C06009.

[28] Krzywda M., Łukasik S., Gandomi A.H.: Graph Neural Networks in Computer Vision – Architectures, Datasets and Common Approaches. In: *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, IEEE, 2022.

[29] Kucharczyk M., Wolter M.: Track Finding with Deep Neural Networks, *Computer Science*, vol. 20(4), 2019. doi: 10.7494/csci.2019.20.4.3376.

[30] LeCun Y., Bengio Y., Hinton G.: Deep Learning, *Nature*, vol. 521, pp. 436–444, 2015. doi: 10.1038/nature14539.

[31] LHCb Experiment, LHCb Collaboration: Event collected at the beginning of 2018 data taking, 2018. http://cds.cern.ch/record/2315673.

[32] Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., *et al.*: PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019. doi: 10.48550/arXiv.1912.01703.

[33] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., *et al.*: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[34] Peterson C.: Track finding with neural networks, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 279(3), pp. 537–545, 1989. doi: 10.1016/0168-9002(89)91300-4.

[35] Piekarczyk M., Bar O., Bibrzycki Ł., Niedźwiecki M., Rzecki K., Stuglik S., Andersen T., *et al.*: CNN-based classifier as an offline trigger for the CREDO experiment, *Sensors*, vol. 21(14), 4804, 2021. doi: 10.3390/s21144804.

[36] PyTorch: MSELoss – PyTorch 1.11.0 documentation. https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html.

[37] Salzburger A.: The ATLAS Track Extrapolation Package, Technical report ATL-SOFT-PUB-2007-005; ATL-COM-SOFT-2007-010, 2007. https://cds.cern.ch/record/1038100.

[38] Scarselli F., Gori M., Tsoi A.C., Hagenbuchner M., Monfardini G.: The graph neural network model, *IEEE Transactions on Neural Networks*, vol. 20(1), pp. 61–80, 2008. doi: 10.1109/TNN.2008.2005605.

[39] Vinyals O., Toshev A., Bengio S., Erhan D.: Show and tell: A neural image caption generator. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, IEEE, 2015.

[40] Zdybał M., Kucharczyk M., Wolter M.: DNN Based Prototype of the Track Reconstruction Algorithm for the MUonE Experiment. In: S.R. González, J.M. Machado, A. González-Briones, J. Wikarek, R. Loukanova, G. Katranas, R. Casado-Vara (eds.), *Distributed Computing and Artificial Intelligence, Volume 2: Special Sessions 18th International Conference*, pp. 202–205, Springer International Publishing, Cham, 2022.

Affiliations

Miłosz Zdybał
The Henryk Niewodniczanski Institute of Nuclear Physics Polish Academy of Sciences, https://www.ifj.edu.pl, milosz.zdybal@ifj.edu.pl

Marcin Kucharczyk
The Henryk Niewodniczanski Institute of Nuclear Physics Polish Academy of Sciences, https://www.ifj.edu.pl, marcin.kucharczyk@ifj.edu.pl

Marcin Wolter
The Henryk Niewodniczanski Institute of Nuclear Physics Polish Academy of Sciences, https://www.ifj.edu.pl, marcin.wolter@ifj.edu.pl

Received: 17.08.2023
Revised: 12.01.2024
Accepted: 12.01.2024

Information for Authors

We accept only the original scientific papers prepared in English. The papers are to be prepared using the LaTeX system. Submitted papers will be refereed by independent reviewers and, if necessary, the Authors may be asked to revise their manuscripts. Proofs will be sent to the Authors for corrections. There is no publication fee. Authors of the accepted papers are eligible to get one hardcopy of the volume containing their contribution free of charge. No postage charges apply.

Our website

https://journals.agh.edu.pl/csci/

Open access

This is an open access journal which means that all content is freely available without charge to the user or his/her institution. This is in accordance with the Budapest Open Access Initiative definition of open access. All printed volumes may be accessed at our website.

Indexing

Google Scholar
http://scholar.google.com

Index Copernicus
http://indexcopernicus.com/

Directory of Open Access Journals
http://www.doaj.org

Open Archives Initiative
http://www.openarchives.org

Digital Libraries Federation
http://fbc.pionier.net.pl/owoc/

BazTech
http://baztech.icm.edu.pl

Worldcat
http://www.worldcat.org

WorldWideScience.org
http://worldwidescience.org

Sherpa Romeo
http://www.sherpa.ac.uk/romeo/

Journal TOCs
http://www.journaltocs.ac.uk

SCOPUS
www.scopus.com

Web of Science – Emerging Sources Citation Index
www.webofknowledge.com

The Computer Science Journal is published by the AGH University of Science and Technology, Krakow Poland. The Editors of the Journal are members of the Faculty of Computer Science, Electronics and Telecommunications and the Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering. The Editorial Board consists of many renowned computer science researchers from all over the world.

The first issue of the Journal was published in 1999. Currently, the Journal is published quarterly, with the main goal to create a forum for exchanging research experience for scientists specialized in different fields of computer science.

Original papers are sought concerning theoretical and applied computer science problems. Example areas of interest are:

- theoretical aspects of computer science,
- pattern recognition and processing,
- evolutionary algorithms,
- neural networks,
- database systems,
- knowledge engineering,
- automatic reasoning,
- computer networks management,
- distributed and grid systems,
- multi-agent systems,
- multimedia systems and computer graphics,
- natural language processing,
- soft-computing,
- embedded systems,
- adaptive algorithms,
- simulation.

Abstracts, full versions of the issued volumes and instructions for authors and reviewers may be found at http://csci.agh.edu.pl