DSIE´19
14th Doctoral Symposium in Informatics Engineering

Faculty of Engineering
University of Porto
Porto | Portugal

Proceedings of the 14th Doctoral Symposium in Informatics Engineering

6 March 2019

Editors:

A. Augusto de Sousa
Carlos Soares

https://paginas.fe.up.pt/~prodei/dsie19/

1st Edition, 2019
ISBN: 978-972-752-243-9
Editors: A. Augusto Sousa and Carlos Soares
Faculty of Engineering of the University of Porto
Rua Dr. Roberto Frias, 4200-465 Porto

DSIE'19 SECRETARIAT:
Faculty of Engineering of the University of Porto
Rua Dr. Roberto Frias, s/n
4200-465 Porto, Portugal
Telephone: +351 22 508 21 34
Fax: +351 22 508 14 43

E-mail: dsie19@fe.up.pt
Symposium Website: https://web.fe.up.pt/ prodei/dsie19/index.html

FOREWORD

STEERING COMMITTEE

DSIE - Doctoral Symposium in Informatics Engineering, now in its 14th Edition, is an opportunity for the PhD students of ProDEI (Doctoral Program in Informatics Engineering of FEUP) and MAP-tel (Doctoral Program in Telecommunications of Universities of Minho, Aveiro and Porto) to show up and prove they are ready for starting their respective theses work.

DSIE is a series of meetings that started in the first edition of ProDEI, in the scholar year 2005/06; its main goal has always been to provide a forum for discussion on, and demonstration of, the practical application of a variety of scientific and technological research issues, particularly in the context of information technology, computer science and computer engineering. DSIE Symposium comes out as a natural conclusion of a mandatory ProDEI course called "Methodologies for Scientific Research" (MSR), this year also available to MAP-tel students, leading to a formal assessment of the PhD students first year's learned competencies on those methodologies.

The above mentioned specific course (MSR) aims at giving students the opportunity to learn the processes, methodologies and best practices related to scientific research, particularly in the referred areas, as well as to improve their own capability to produce adequate scientific texts. With a mixed format based on a few theoretical lessons on the meaning of a scientific approach to knowledge, practical works on analysing and discussing published papers, together with multidisciplinary seminars, the course culminates with the realization of a DSIE meeting. Then, DSIE may be understood as a kind of laboratory test for the concepts learned by students. In this scope, students are expected to simultaneously play different roles, such as authors of the submitted articles, as members of both organization and scientific committees, and as reviewers, duly guided by senior lecturers and professors.

DSIE event is then seen as the opportunity for the students to be exposed to all facets of a scientific meeting associated with relevant research activities in the above mentioned areas. Although still at an embryonic stage, and despite some of the papers still lack of maturity, we already can find some interesting research work or promising perspectives about future work in the students' thesis. At this moment, it is not yet essential, nor often possible, for most of the students in the first semester of their PhD, to produce sound and deep research results. However, we hope that the basic requirements for publishing an acceptable scientific paper have been fulfilled.

Each year DSIE Proceedings include papers addressing different topics according to the current students' interest in Informatics. This year, the tendency is on Software Engineering and Robotics, Artificial Intelligence and Machine Learning, Text Mining and

Telecommunications.

The complete DSIE'18 meeting lasts one entire day and includes one invited talk by an academic researcher.

Professors responsible for ProDEI program's current edition, are proud to participate in DSIE'19 meeting and would like to acknowledge all the students who have been deeply involved in the success of this event. Hopefully, this involvement contributes for their better understanding of the themes addressed during the MSR course, the best scientific research methods and the good practices for writing scientific papers and conveying novel ideas.

Porto, March 2019
Carlos Soares and A. Augusto de Sousa
(Steering Committee of DSIE 2019)

FOREWORD

ORGANIZING AND SCIENTIFIC COMMITTEES

The chairs of the Organizing and Scientific Committees of the Doctoral Symposium in Informatics Engineering (DSIE'19) warmly welcome you to the DSIE 14th edition. With a great honour, we have accepted the invitation to be a part of these committees. Organizing an event, like the DSIE, confirmed to be both a challenging and practical task, in which all the persons involved have certainly derived great value.

The joint effort of our colleagues from the Doctoral Programme in Informatics Engineering (ProDEI) and the Doctoral Programme in Telecommunications (MAP-tele), was instrumental in making this event a success. We believe that these efforts are reflected in the quality of the communications realized and the organization in general.

Our first acknowledgment goes to our supervisors, Professor Augusto Sousa and Professor Carlos Soares. We would like to thank them for their time and their efforts in making this conference possible and for providing us with all invaluable concepts.

We would like to thank all the senior members of the Scientific Committee for their involvement, the junior members for their collaboration, and the invaluable support of Sandra Reis and Pedro Silva (DEI) from the Informatics Engineering Department of Faculty of Engineering - University of Porto.

And, above all, we thank you for being a part of DSIE'19!

Porto, March 2019
Luis Roque and Paula Silva (Scientific Committee Chairs)
Mafalda Falcão and Pedro Peixoto (Organization Committee Chairs)

CONFERENCE COMMITTEES

STEERING COMMITTEE
A. Augusto Sousa
Carlos Soares

ORGANIZING COMMITTEE CHAIR
Mafalda Falcão Ferreira
Pedro Peixoto

ORGANIZING COMMITTEE
David Freitas
Ferenc Tamási
Flávio Couto
Hajar Baghcheband
Leonardo Ferreira
Luis Roque
Miguel Abreu
Paula Silva
Pedro Peixoto

SCIENTIFIC COMMITTEE CO-CHAIRS
Luis Roque
Paula Silva

SENIOR SCIENTIFIC COMMITTEE
Ali Shoker
Ana Rocha
Aníbal Ferreira
Carla Lopes
Carlos Ferreira
Daniel Silva
Filipe Correia
Gil Gonçalves
Henrique Cardoso
Henrique Salgado

Hugo Ferreira
João Bispo
João Faria
João Moreira
Luís Reis
Luís Teixeira
Rui Abreu
Rui Rodrigues
Sérgio Nunes

JUNIOR SCIENTIFIC COMMITTEE
Amir Farzamiyan
André Coelho
David Freitas
Ehsan Shahri
Ferenc Tamási
Flávio Couto
Hajar Baghcheband
Leonardo Ferreira
Luís Roque
Mafalda Falcão
Miguel Abreu
Paula Silva
Pedro Peixoto

SPONSORS

DSIE'19 – Doctoral Symposium in Informatics Engineering is sponsored by:

Contents

INVITED SPEAKER

ADÉRITO FERNANDES MARCOS

Adérito Fernandes-Marcos is Full Professor at Aberta University (the Portuguese public open distance learning university). He is founder and director of the Doctoral Program in Digital Media Art, running since 2012 in e-learning mode, offered in association by Aberta University and University of Algarve. He is an integrated member of the Research Centre for Arts and Communication; and research collaborator of INESC TEC – Institute for Systems and Computer Engineering, Technology and Science. Dr. Fernandes-Marcos is founder and President of the Artech-International – International Association of Computer Art. He is editor-in-chief of the International Journal of Creative Interfaces and Computer Graphics (ISSN: 1947-3117); and of the novel ART(e)FACT(o) – International Journal of Transdisciplinary Studies on Artefacts in Arts, Technology and Society (ISSN: 2184-2086).

SESSION 1

Software Engineering and Robotics

Experimental Evaluation of Formal Software Development Using Dependently Typed Languages

Ferenc Tarnási
up201809113@fe.up.pt
Faculty of Engineering, University of Porto

***Abstract*—We will evaluate three dependently typed languages, and their supporting tools and libraries, by implementing the same tasks in each language. One task will demonstrate the basic dependent type support of each language, the other task will show how to do basic imperative programming combined with theorem proving, to ensure both resource safety and functional correctness.**

***Index Terms*—formal software development, dependent types, Coq, Iris, Agda, Fstar, ST monad, Hoare monad, Dijkstra monad**

I. Introduction

Dependently typed programming is getting some attention in the past years. Noticeable for instance in [2], where prominent researchers in the area state that *"Dependently typed programming languages like Agda are gaining in popularity, and dependently typed programming is also becoming more popular in the Coq community, for instance through the use of some recent extensions."*

The interest is motivated by the need to find the right balance between usability and flexibility when applying the increased accuracy of dependent types in describing program behavior. One can use dependent types in situations ranging from disciplined dynamic typing (Dependent JavaScript [18]), to prove memory correctness of the standard library of a statically typed language (RustBelt [27]), or correctness of a compiler (CompCert [31]).

This growing popularity is also demonstrated by the active tool development to explore working with dependent types, Wikipedia lists 11 actively developed languages with dependent type support.

Three of the tools used in academia are compared, in the context of formal software development. We describe the performed experiment of executing the same tasks in the selected environments.

This paper is structured as follows: in Section II we provide an introduction of dependent types. In Section III, we describe the tool selection process, and a short introduction of the selected tools. In Section IV, we describe the experiments that will be conducted with each of the selected tools. In Section V we describe the implementations of each of the selected tasks with each of the selected tools and explain the experimental results that we have obtained. Finally, in Section VI the conclusions drawn from our experiments are presented, as well as possible future areas of interest.

II. Background

Dependently typed languages [39] extend traditional typed languages, by allowing the types of values to depend on other values. For illustration purposes, we will introduce the concept of dependent types in pseudo C++. *

In standard C++, it is possible to define the type `DepType` as seen in Listing 1, but the `value` template parameter must be available at compile time.

```
class DepType<int value> {};
```

Listing 1: Compile time dependent type in C++

If C++ would be a dependently typed language, we could define the function `pi()` as shown in Listing 2 where the argument is only available at runtime, but the return type depends on the argument's value. Another, more explicit example is the function `pi2()` depicted in Listing 2, where the return types are not versions of the same base type. In `div2` a simple constraint is presented, ensuring that the function can only be called if the returned values are exactly one half of the argument.

```
auto pi(int x) -> DepType<x> {
  return DepType<x>();
}
// return type depends on runtime value of 'x'
auto pi2(int x) -> (x ? int : (char const *))
    {
  return x ? 1 : "Hello World!";
}
// only allow calls, if 'x' is even
int div2(int x, (x % 2 == 0 ? std::monostate :
     void) x_is_even) {
  return x / 2;
}
// property helper: mapping booleans to types.
// true is a trivially produceable value
// false is a non-produceable value
#define Prop(e) (((e) ? std::monostate : void)
    )
```

Listing 2: C++11 II

And we could define a `struct` like various `Sigma...` in Listing 3. Here a dependently typed C++ could check, that values of type `Sigma_dependent_type` can only be

*A similar attempt for refinement types can be found in [48] and [44].

created if the constraint described in the type of `i_or_s` is satisfied.

A Sigma (Σ) type (also called dependent pair) in a dependently typed language is a structure with two elements, where the *type* of the second element depends on the `value` of the first element.

```
struct Sigma_class {
  int x;
  DepType<x> d;
};

struct Sigma_union {
  int x;
  union {
    short i; // x != 0
    char const * s; // x == 0
  };
};

struct Sigma_dependent_type {
  int x;
  (x != 0 ?
    short :
    char const *
  ) i_or_s;
};
```

Listing 3: C++ Σ

This idea allows us to describe properties of values. In Listing 4, a type is defined, which can only hold even integers. The assurance of evenness of `x` depends on the impossibility of creating a value of type `void`, which is the calculated type of the field `evenness_proof` in the odd `x` case. In case of an even `x`, the calculated type for the proof field is `std::monostate`, a type that has a single possible value, thus has no information content.

We could use any other type instead of `monostate`, but this expresses the intent that we don't care what the value is, as long as it exists (as opposed to the `void` case, where we want to ensure non-existence).

```
struct even_ints {
  int x;
  (x % 2 == 0 ? std::monostate : void)
      evenness_proof;
};
```

Listing 4: Even integers

This sort of constrained types are called refinement types [22] their general form in our pseudo dependent C++ is shown in Listing 5. (Which itself is a specialized form of the Σ types from Listing 3).

```
template <typename T, bool (*P)(T)>
struct refined_T {
        T v;
        (P v ? std::monostate : void) proof;
};
```

Listing 5: Dependent C++ refinement types

The analogy of "a `value` to its *type*, is what a `proof` is to it's *logical formula*", is described as the Curry-Howard correspondence [26]. In our pseudo C++ the type of the proofs are always either `void` or `std::monostate`, depending on the condition's value in the ternary expression. In proper dependently typed languages on the other hand, logical formulas are themselves types. For example, a conjunction of two formulas is the type, that has two type parameters, and in order to create a value (that is a proof of the conjunction), one has to provide two values, each with a corresponding type.

```
template <typename A, typename B>
struct Conjunction {
        Conjunction(A, B) {}
};
template <typename A, typename B>
struct Disjunction {
        static Disjunction left(A a) {}
        static Disjunction right(B b) {}
};

/* using the property helper macro from above,
   a refinement type expressing, that an
       integer
   is within the specified range. */
template<int low, int high>
struct Range {
        int x;
        Conjunction<Prop(low <= x), Prop(x <
            high)> proof;
};
```

Listing 6: Dependent C++ logical connectives

For a more complete illustration Listing 8 a sorted linked list implementation is shown in dependent C++.

```
struct List {
  int v;
  List * next;
};
/* struct sortedlist: a type expressing that a
     List, starting
  from node, is sorted */
struct SortedList {
  List * node;
  SortedList * nextproof;
  /* property type expressing sortedness,
    depends on values: node and nextproof. */
  Disjunction<
    Prop(node == nullptr),
    Disjunction<
      Prop(node->next == nullptr),
      Conjunction<
        Prop(node->v <= node->next->v),
        Conjunction<
          Prop(nextproof != nullptr),
          Prop(nextproof->node = node->next)
        >
      >
    >
  > proof_value; // the proof_value field is
      compile time only
};
```

Listing 7: C++ sorted list

Building up proof terms is similar to calculating traditional values, see in Listing 8 as we build up `proof_value`.

```
SortedList *
prepend(
  int v,
  SortedList * l,
  Prop(l == nullptr || v <= l->node->v)
      v_proof
)
{
  List * node = new List{v: v, next: l ? l->
      node : nullptr};
  SortedList * res = new SortedList{
    node: node,
    nextproof: l,
    proof_value: node == nullptr ? Disjunction
        <...>::left(...) : Disjunction<...>::
        right(...)
    /* the developer builds up a value of the
        type
       specified above, and the compiler
       checks the validity of the types */
  };
}
```

Listing 8: C++ sorted list prepend

III. Tool Selection

The search term `TITLE-ABS-KEY("dependent* type*" AND imperative)` returned 40 hits at Scopus, of those papers 25 are unique. From the unique papers, we selected those that were not introducing a language, but using the language as a tool, in order to select languages that the community found useful in research.

This selection criteria resulted in Agda [51] used in [1], Coq [6] used in [37], [24], and [46], and F* [8] used in [11] [12].

A. Selected Tool Short Introduction

a) Agda

Agda is the name of both the dependently typed functional programming language, and the interactive proof assistant to work with the language, based on typed holes [41] implemented as an Emacs mode.

Agda is based on intuitionistic type theory [38], a foundational system for constructive mathematics. We examined version 2.5.4.1, with stdlib version 0.17.

b) Coq

Coq is the name of a proof management system. It is built on three languages, *Gallina* the dependently typed functional language, *Vernacular* the proof engine management language, and L_{tac} the language for proof tactics. There are multiple interactive environments developed for Coq, the official is CoqIDE, but ProofGeneral for Emacs is also popular.

Coq is also based on intuitionistic type theory. We examined version 8.8.2.

*c) F**

F* (pronounced F star, sometimes written as F$\star$) is a general-purpose functional programming language, with support for program verification, based on dependent types. F* though supports dependent types, it is mainly focused on the refinement type subset.

F* does not make a statement about its foundational logic. We examined version 0.9.6.0.

B. Quick Comparison

The selected tools are all based on languages that support dependent types. The syntax of each language is described in the following resources: Agda [40], Coq [7], and F* [9].

To get an overview of how each language looks like, in the three listings below the same function is defined three times. The function takes a natural number as a parameter and returns a dependent pair as a result.

The result pair's first element in the function body is always set to zero (this is to simplify the example), and the type of the result pair's second element depends on both the function parameter's value (`x`), and the pair's first element's value (`y`). (Since we always set the first element to zero, this is effectively a comparison of the function parameter with zero). The second element's type is either the unit type, or boolean, depending on the comparison result.

Agda:

```
open import Data.Bool using (Bool;
    if_then_else_)
open import Data.Nat using (ℕ; suc; zero; _≟_)
open import Data.Product using (∃; _,_)
open import Data.Unit using (⊤)
open import Relation.Nullary.Decidable using (
    ⌊_⌋)

pi : (x : ℕ) → ∃ (λ (y : ℕ) → (if ⌊ x ≟ y ⌋
    then ⊤ else Bool))
pi zero = (zero , ⊤.tt)
pi (suc _) = (zero , Bool.true)
```

Coq:

```
Require PeanoNat.

Definition pi (x : nat) : { y : nat & if
    PeanoNat.Nat.eq_dec x y then unit else
    bool } :=
  match x return { y : nat & if PeanoNat.Nat.
      eq_dec x y then unit else bool } with
  | O => existT _ O tt
  | S x' => existT _ O true
  end.
```

F*:

```
module PiSigma

val pi : x:nat -> Tot (y:nat & (if x = y then
    unit else bool))
let pi x =
  match x with
```

```
  | 0 -> (|0, ()|)
  | _ -> (|0, true|)
```

Listing 9: ΠΣ in three languages

From this short syntax comparison it is already visible, that the tools take different approaches: in Agda we need to import even the most basic definitions, while in F* we don't need to import anything; Agda typically uses Unicode symbols, while the others use ASCII names. This is only a convention of the developers of the tools, as both F* and Coq has the ability to work with Unicode characters. A library for Coq called Iris [28] for example employs Unicode extensively.

IV. Tasks

We selected two tasks to implement, that represent two areas of functionality:

A. Theorem Proving

Prove the commutativity of addition over the language's default natural ($\mathbb{N}$) type*. That is, for all $a, b \in \mathbb{N}$, the equality $a + b = b + a$ holds. This task exercises the basic theorem proving machinery in the language.

B. Imperative Programming using In Memory Datastructures

Sort an in memory array of fixed size integers. This task demonstrates the language's prowess in combining safe memory management and proving application level properties [52].

Ensuring valid memory addressing is one important use case of dependent types. This problem is mostly mitigated by the hardware getting fast enough to afford runtime bounds checks, and the compilers getting clever enough to elide most of the runtime bound checks†. So this task aims to demonstrate the other important feature of dependent types: the ability to describe high level requirements and certify their implementation (in this case sortedness).

V. Implementation and Results

A. Theorem Proving

1) Agda

a) getting started

Agda is popular enough, that an internet search led us to a partial solution of this problem ‡.

As Agda does not autoload even the most basic definitions, it takes some time to discover, where a definition is located in the standard library. Also, if one wishes to write idiomatic Agda, and the location of a definition is not the canonical way to import a symbol, one has to chase down the wrapping module, that imports, then re-exports the original definition.

b) ergonomics

Agda has an Emacs mode §, where one can use a hole based development style. To create a hole, one enters a question mark (`?`) in place of an expression. The editor then creates a hole context, in which the developer can interactively build up the expression with type the hole requests.

The hole context provides an overview of what values of what types are available, and what is the type of the expression the developer needs to create.

2) Coq

a) getting started

Since the author is quite familiar with Coq already, we had to try to rely on intuition and memory to try to evaluate the starting out experience of Coq.

Coq has a very steep learning curve, but since it is a very mature project, there are plenty of tutorials online, and the tooling is rather featureful and stable.

A similar problem to Agda of standard library discoverability exists in Coq as well, but the situation is improved by the integrated `Search` commands ¶, which find in the current context facts about types or functions.

b) ergonomics

Coq is the tool of the three reviewed, that has the most mature proof facilities.

Coq is designed around interactive proof development, which is similar to the hole based approach of Agda, but it not only provides the context for the developer, but also gives tools to transform the goal and the available values in the context.

When using the interactive facilities, the author proceeds, and issues tactics ‖, that transform the hole, introduce new facts to the context, split the target into parts, for a full list see the Coq Reference Manual.

The implementation presented in `PlusComm.v` is written in the interactive proving style.

c) non-interactive proving

To provide a more direct comparison, we proved the commutativity in the direct style of Agda and F* in `PlusCommDirect.v`. This leads to a very similar proof as with the other tools. One gives a fully formed proof to the language for checking, with no help from the tool.

3) F*

a) getting started

Simple proofs like this can be discharged with the integrated Z3 [19] satisfiability modulo theories solver. F* by convention uses refinement types, in particular the refinement of the unit type, to represent properties. F*, though does not encourage it, is also able to express the original properties-are-types idea of dependent types.

b) ergonomics

F* also has an Emacs mode, that is the recommended way of editing F* sources, called fstar-mode **. It is still in

The default varies between languages, in F it is a refined type, limiting a base type to non-negative values, in Agda and Coq it is a Peano numeral.

†See for example Java, Python, or Rust.

‡https://stackoverflow.com/questions/52282786/proving-commutativity-of-addition-in-agda

§`elpa-agda2-mode` package in Debian

¶https://coq.inria.fr/refman/proof-engine/vernacular-commands.html#coq:cmd.search

‖https://coq.inria.fr/distrib/current/refman/coq-tacindex.html

**https://github.com/FStarLang/fstar-mode.el

early development phase, so some features are not working. Most problematic is the environment's reluctance to work with incomplete source, which is quite the common occurrence during programming.

One useful technique to deal with this limitation is using the `admit()` function in place of the missing expressions in the code. This is similar to Agda's hole oriented programming, but it does not provide the helpful interactive context, but helps with the partial source problem.

c) F task with Peano numbers*

Since the task following the original description was so quickly and smoothly solved by F*, we decided to include the task implemented for Peano [42] numbers, using both the refined-unit-as-prop approach in `PlusCommPeano.fst`, and an explicit type-as-prop in `PlusCommPeanoProp.fst`. During the implementation of these solutions, the immaturity of the proof development environment forcing us to provide the solutions without support of the tool was a little bothering, but peeking at the Agda solution helped the proof along.

The solution in `PlusCommPeano.fst` still relies on the built-in Z3 automation. Since we are not using the built in numerical types the proof itself shows a little more of the internals.

The solution in `PlusCommPeanoProp.fst` is managing the proof terms explicitly, and it seems this method of proving disables the built-in proof automation, as the full proof term had to be entered.

Even though F* supports proof automation through tactics, since these are not interactive, they don't help when such a small scale task is developed. But we expect, in more complicated tasks (e.g. in a domain specific language implemented in F*), they can be quite useful.

B. Imperative Programming using In Memory Datastructures

1) ST&Hoare introduction

One way of handling stateful computation is through the ST monad [35] introduced in Haskell. The ST monad provides primitives to work with the heap, but it prevents direct access to the memory. In fact the ST monad, effectively hides the values that are in memory from the host language. The established way to workaround this, is to use Hoare logic [25].

In the Hoare monad the ST monad is enriched with pre- and post-conditions around ST operations. This enriched construct is called the Hoare triple. It consists of: the *pre-condition*, which specifies the requirements about the environment for when the *action* is enabled; the ST *action* which defines the operation to be performed; and the *post-condition*, which specifies the guarantees after the ST operation is performed, based on the values in the heap both before and after the operation, as well as the value generated by the ST operation.

A newer structure, called the Dijkstra monad [47], is also used, which fulfills the same function as the Hoare monad, but instead of pre- and post-conditions, it uses weakest precondition predicate transformation [20]. A weakest precondition (WP) predicate transformer generates a pre-condition, based on a post-condition, that is the least restrictive pre-condition, that enables the execution of the ST action.

2) Agda

a) ST in Agda

Unfortunately Agda does not include the ST monad in the standard distribution, so we used an implementation from [32]. In [32] Kovács models what in Haskell is called STRef [10], but limiting the supported types to boolean and natural numbers. It doesn't support monadic bind operation either, so we had to resort to continuation passing [5]

b) modal logic

Since in order to reason about the changes in the ST heap, we would need some sort of modal logic [36] over the values stored in the heap (to be able to talk about before/after values). But Hoare logic is not included in the implementation of Kovács's ST implementation, so we abandoned the attempt of proving the sortedness of the resulting list.

We settled for only showing how to work with memory in the imperative style, and only giving guarantees about the validity of indexing in the array (we could do this, since the indexing happens in the host language), not about the sortedness of the result (which would require access to the values stored in heap memory).

3) Coq

a) ST in Coq

Coq does not include imperative features in its standard library. Since Ynot [17] was used in [24] as the library implementing mutable state, we first tried to use that, but we found, that it has been abandoned since 2014, and does not compile with the latest Coq. An actively developed similar library for Coq is Iris [28]. We examined Iris development version with Git hash 455fec93.

Iris has a larger scope, namely it also targets concurrent programs, but in contrast to Ynot, Iris does not support compiling the program to executable format (called extraction in Coq [*]). This follows from the fact, that Ynot uses shallow embedding and Iris uses deep embedding.

Both Ynot and Iris weaken the Coq guarantees, by introducing the possibility of creating non terminating programs, which are disallowed by vanilla Coq.

b) modal logic

Iris is based on concurrent separation logic [33] we will use the instantiation of the base logic for memory heaps. The implementation uses the Dijkstra [47] monad is based on weakest precondition transformation [20], as opposed to the Hoare monad, that is based on pre- and post-conditions [25]. In practice, since the pre- and post-conditions are more natural to think about, the predicate transformers of weakest precondition calculus is hidden from the developer, and the predicate transformer is generated from the provided pre- and post-conditions.

c) ghost variables

Iris uses ghost variables to help express properties of the program. Ghost variables *can not* interact with the evaluation

[*]https://coq.inria.fr/distrib/current/refman/addendum/extraction.html

of the program, they are only present while proving program properties.

The ghost variable is connected to the real variables through properties. It is said, that a ghost variable models a real variable. For example in this task, we are modeling an array, using a pointer as real variable, and a list as ghost variable, expressing, that the pointer points to a value that is equal to the value of the first element of the list, the (pointer+1) points to a value that is equal to the second element of the list, $\forall i : \mathbb{N}, i < |list| \implies pointer + i \mapsto list!!i$

d) proof management

Coq itself is an interactive proof assistant, so the basic mode of operation is building proofs, by interactively applying tactics that transform the goal [*].

Coq also supports proof automation, which involves automated proof term generation, and proof search for fitting terms. Iris uses this facility and the typeclass system of Coq extensively, creating a fourth and fifth language on, top of the three languages already in Coq. **Iris logic**, a DSL implementing an affine Concurrent Separation Logic (CSL) [13] . And **Iris Proof Mode**, a tactic language to deal with proofs in Iris logic [34].

e) discoverability

Coq itself is well established and well documented, with many tutorials to choose from [†].

Iris on the other hand is still under active development (2.0 released in 2016, 3.0 in 2017), finding the relevant documentation is challenging, and sometimes the relevant documentation does not exist (c.f: [29] chapter 1.3).

f) location arithmetic

The base logic does not define arithmetic operations for locations (pointers), so for demonstration purposes we added an indexing extension `location_arithmetic`.

A proper location arithmetic should take into account the size of the allocation, but for simplicity, we defined an array as elements separated by one "unit" of whatever an increment of a location value by one means, as this is not material to the meaning of the proof , but simplifies the proofs themselves.

g) numeric conversions

Locations are represented as `positive` ($\mathbb{Z}^+$) numbers, the standard library mostly uses $\mathbb{N}$, and the default number type is $\mathbb{Z}$.

The interaction of these three number types creates a huge time sink, as the usual rules of mathematics do not apply anymore. The built in conversion from `nat` ($\mathbb{N}$) type maps $0_{\mathbb{N}}$ to $1_{\mathbb{Z}^+}$. This means, for example, that depending on whether we convert the arguments, or the result of an addition, we get different results: $(0_{\mathbb{N}} +_{\mathbb{N}} 0_{\mathbb{N}})_{\mathbb{Z}^+} \neq (0_{\mathbb{N}})_{\mathbb{Z}^+} +_{\mathbb{Z}^+} (0_{\mathbb{N}})_{\mathbb{Z}^+}$, the left side is 1 the right side is 2.

h) fun with separation logic

Separation logic is a mixture of linear and nonlinear logic [3], which for us means, that facts about a variable in the liner logic part can only be used once.

Proving with separation logic used by Iris compared to the standard intuitionistic logic used by Coq, forces a more disciplined approach to proving, which at first does pose some difficulties, but it also helps offload some mental burden from the developer to the compiler [14].

In this task keeping track of memory resources is solved by the affine logic perfectly (as it was designed to do). [‡]

i) predetermined heap types

The CSL DSL only supports a pre-determined list of types [§]. This limited the sorting predicates as well, since only the operators in the DSL can be used.

j) proof length

As we get to higher level operations, the associated proofs get shorter, this gives a probable explanation, why large projects use Coq (RustBelt [27], CompCert [31], Fiat-Crypto [21], VST [4]).

*4) F**

*a) ST in F**

F* uses algebraic effects [43] for modeling stateful computations. F* implements the ST effect in its standard library.

b) discoverability

We found the discoverability of the F* libraries lacking, but once we settled to base the implementation on `examples/algorithms/QuickSort.Array.fst` from the F* source distribution, the standard library turned out very well equipped to deal with sorting.

c) modal logic

F* also uses the Dijkstra monad [47] to keep track of the programs environment like Coq+Iris.

d) proof management

F* relies on implicit proofs, generated from the provided preconditions proving the post-conditions. This makes the proof process opaque, and in case the goals are not discharged, an exercise in guessing what the automated proof machinery wants as input, to be able to find the solution, and which knobs of the magic machine has to be tweaked to help it through the proof search.

The approach we took was, to throw more and more facts at the proof search, and once it succeeds, start removing the ones, that keep the goals discharged. Not a very efficient, scalable, or dignified way to work. But alas no alternative exists, barring one becomes intimately familiar with the internal proof searching algorithms of F*. Then repeat the exercise for the next F* releases, ad infinitum.

e) array lib

F*'s array implementation [¶] can not track individual cell modifications with the `modifies` utility of ST, only the whole array can be declared as modified with the `modifies` keyword.

[*]The whack-a-mole style proving http://gallium.inria.fr/blog/coq-eval/

[†]https://coq.inria.fr/documentation

[‡]Brady in [16] demonstrates the usefulness of linear logic (a slightly stricter version of affine logic) in the context of **Id**ris. Brady presenting this can be found here: https://www.youtube.com/watch?v=mOtKD7ml0NU&t=30m53s)

[§]boolean, $\mathbb{Z}$, unit, location (pointer), prophecy (which seems to be an internal type)

[¶]`FStar.Array`

C. Quantitative Analysis

a) source metrics

In Table I we are showing a numerical evaluation of the size of our solutions. Columns task 1 and task 2 refer to the number of lines in the solution for task 1 and 2 respectively, counting all non-comment lines.

In columns body 1 and body 2, we show the number of lines, with comments and import statements removed, while in core 2 we show the number of lines directly related to sorting and the sortedness proof, excluding the generic proofs that should be added to either the standard library of the tool, or the array library.

TABLE I: Number of lines per task per tool

	task 1	body 1	task 2	body 2	core 2
Agda	48	45	123*	80*	26*
Coq	12	12	1433	1265	574
F*	3	3	109	90	65

*: The agda solution for the second task only proves memory safety, not sortedness.

Agda has the biggest overhead associated with library imports (the difference between the task and body columns are 18% and 30%). This is a blessing and a curse at the same time, as it makes writing the code more tedious, but the one reading the code is helped by the explicit dependency enumeration.

In absolute numbers the Coq solution is an order of magnitude larger than the other two solutions. This is only a fair comparison against F*, but it still shows that proof search in F* does work*.

b) development time

The rough approximation of the time to solve tasks 1&2: Agda 6 days, Coq 7 days, F* 3 days. These numbers are based on the version control history of the author. Here too, Coq is the one requiring the most time, even though the author has the most experience with it.

VI. Conclusion

The three languages take very different approaches to present the power of dependent types to the user. Thus it is impossible to declare a best tool, but we will describe the situation in which each tool excels.

a) history

Coq is the oldest of the three, with many successful industrial [27] [31] and academic [23] projects under its belt.

Agda is also established, especially in programming language research [2] [30].

F*, a relatively recent development coming from Microsoft, discourages creating proof terms by hand, presumably to appeal to users who wish to avoid dealing with the minutia of proofs. This is great as long as the user can stay within the confines of the F* design, but at the cost of a sudden increase in discomfort, if one must leave the beaten path.

Based on their history, Coq can be considered the standard tool, when one wishes to work with something that "everybody else" uses, and a tool that will probably be around later.

b) proving

Coq uses interactive tactics to prove goals, which is very convenient, but may lead to large proof scripts in case one does ad-hoc proofs. But Coq also has the tools to make the proofs concise, provided one works in a fixed domain, and creates the necessary abstractions. Iris for example embedded the full logic of CSL and tactics to work with it in Coq.

Agda is more in line with traditional programming languages, as it expects the user to write the expressions that will produce the expected value.

F* does not want the user to prove anything, it only expects enough facts to be presented, so that the built-in prover can work out a proof.

The choice of tool depends very much on the task one wishes to solve. F* works great, as long as one can fit the task at hand into what F* can work with, and one is willing to do the guesswork involved in trying to work out what the missing piece might be for the automated prover to go through.

Coq and Agda both provide interactive proving environments, but the larger user base and longer history of Coq give an edge that Agda can't compete with.

c) messy requirements vs messy proofs

As we stated in the previous paragraph, F* discourages user proofs, but this makes requirements unnecessarily large, since the automated tool needs a lot more detail, than what a human prover needs to prove the same goal.

If one can fit one's work in F*'s beaten path, then it works great, otherwise Coq or Agda is probably a better choice as they provide a more natural environment to create proof terms.

d) tactics generated vs hand crafted proofs

In [50] Wadler states *"Proofs in Coq require an interactive environment to be understood, while proofs in Agda can be read on the page."*, while this is true for the languages themselves, but Proviola [49] can alleviate this problem of Coq, by recording the proof state after each tactic execution, and producing an html document with the proof state added for each tactic. F* does not have this problem, as the proof terms do not appear either in the source, or during proving.

Whether it is easier to read complete proof terms, or the replay of a step by step creation of a proof term is dependent of the task at hand, but the author thinks, that it is more straightforward to create scripts step by step in Coq, though it does require discipline on the programmer's part, so as to not create a write-only script †.

1) future work

Both the breadth and the depth of this work could be extended. Doing the same tasks in other, less established languages like Idris [15], or ATS [53], or trying different libraries like FCSL [45].

The depth increased by adding more interesting tasks, for example, investigating the generation of verified executables

At least for this case, after appeasing the search machinery. It would be interesting to see in a larger project, with not so straightforward properties, how would the built in logic of F behave?

†http://www.jargon.net/jargonfile/w/write-onlylanguage.html

from the verified sources, or comparing how different tools enable verifying resource management other than memory (files, network sockets, etc), or verifying non functional requirements like security or real time constraints.

REFERENCES

[1] Stephan Adelsberger, Anton Setzer, and Eric Walkingshaw. Developing gui applications in a verified setting. Cham, 2018.

[2] Thorsten Altenkirch, Nils Anders Danielsson, Andres Löh, and Nicolas Oury. $\pi\sigma$: Dependent types without the sugar. In *International Symposium on Functional and Logic Programming*, 2010.

[3] Andrew W Appel. Tactics for separation logic. *INRIA Rocquencourt and Princeton University, Early Draft*, 2006.

[4] Andrew W Appel. *Program logics for certified compilers*. Cambridge University Press, 2014.

[5] Andrew W Appel and Trevor Jim. Continuation-passing, closure-passing style. In *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 293–302. ACM, 1989.

[6] Coq Authors. https://coq.inria.fr/, 2018.

[7] Coq Authors. Documentation — the coq proof assistant. https://coq.inria.fr/documentation, 2018.

[8] F* Authors. https://fstar-lang.org/, 2018.

[9] F* Authors. F* tutorial. https://www.fstar-lang.org/tutorial/, 2018.

[10] Haskell Wiki Authors. Data.STRef. https://hackage.haskell.org/package/base-4.12.0.0/docs/Data-STRef.html.

[11] Karthikeyan Bhargavan et al. Everest: Towards a Verified, Drop-in Replacement of HTTPS. In *n/a*, Dagstuhl, Germany, 2017.

[12] Karthikeyan Bhargavan, Cedric Fournet, and Markulf Kohlweiss. mitls: Verifying protocol implementations against real-world attacks. *IEEE Security & Privacy*, 14(6):18–25, 2016.

[13] Lars Birkedal and Aleš Bizjak. *Lecture Notes on Iris: Higher-Order Concurrent Separation Logic*. 2018.

[14] Rúnar Bjarnason. Maximally powerful, minimally useful. http://blog.higher-order.com/blog/2014/12/21/maximally-powerful/, 2014.

[15] Edwin Brady. Idris, a general-purpose dependently typed programming language: Design and implementation. https://www.idris-lang.org/, 2013.

[16] Edwin Brady. *Type-driven development with Idris*. Manning Publications Company, 2017.

[17] Adam Chlipala, Gregory Malecha, Greg Morrisett, Avraham Shinnar, and Ryan Wisnesky. Effective interactive proofs for higher-order imperative programs. *ACM Sigplan Notices*, 44(9):79–90, 2009.

[18] Ravi Chugh, David Herman, and Ranjit Jhala. Dependent types for javascript. *ACM SIGPLAN Notices*, 47(10):587–606, 2012.

[19] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.

[20] Edsger Wybe Dijkstra. *A discipline of programming*, volume 1. prentice-hall Englewood Cliffs, 1976.

[21] Andres Erbsen. *Crafting certified elliptic curve cryptography implementations in Coq*. PhD thesis, Massachusetts Institute of Technology, 2017.

[22] Tim Freeman. Refinement types ml. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1994.

[23] Georges Gonthier. A computer-checked proof of the four colour theorem, 2005.

[24] Colin S Gordon, Michael D Ernst, and Dan Grossman. Rely-guarantee references for refinement types over aliased mutable data. In *ACM SIGPLAN Notices*, volume 48, pages 73–84. ACM, 2013.

[25] Charles Antony Richard Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.

[26] William A Howard. The formulae-as-types notion of construction. *To HB Curry: essays on combinatory logic, lambda calculus and formalism*, 44:479–490, 1980.

[27] Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, and Derek Dreyer. RustBelt: Securing the foundations of the rust programming language. 2017.

[28] Ralf Jung, Robbert Krebbers, Lars Birkedal, and Derek Dreyer. Higher-order ghost state. In *ACM SIGPLAN Notices*, volume 51, pages 256–269. ACM, 2016.

[29] Ralf Jung, Robbert Krebbers, Jacques-Henri Jourdan, Aleš Bizjak, Lars Birkedal, and Derek Dreyer. Iris from the ground up: A modular foundation for higher-order concurrent separation logic. *Journal of Functional Programming*, 28, 2018.

[30] Wolfram Kahl. Dependently-typed formalisation of relation-algebraic abstractions. In *International Conference on Relational and Algebraic Methods in Computer Science*, pages 230–247. Springer, 2011.

[31] Daniel Kästner, Xavier Leroy, Sandrine Blazy, Bernhard Schommer, Michael Schmidt, and Christian Ferdinand. Closing the gap – the formally verified optimizing compiler CompCert. CreateSpace, 2017.

[32] András Kovács. Computing ST monad in vanilla Agda. https://gist.githubusercontent.com/AndrasKovacs/07310be00e2a1bb9e94d7c8dbd1dced6.

[33] Robbert Krebbers, Ralf Jung, Aleš Bizjak, Jacques-Henri Jourdan, Derek Dreyer, and Lars Birkedal. The essence of higher-order concurrent separation logic. In *European Symposium on Programming*, pages 696–723. Springer, 2017.

[34] Robbert Krebbers, Amin Timany, and Lars Birkedal. Interactive proofs in higher-order concurrent separation logic. *ACM SIGPLAN Notices*, 52(1):205–217, 2017.

[35] John Launchbury and Simon L Peyton Jones. Lazy functional state threads. In *ACM SIGPLAN Notices*, volume 29, pages 24–35. ACM, 1994.

[36] Clarence Irving Lewis and Cooper Harold Langford. *Symbolic logic*. 1932.

[37] Gregory Malecha, Greg Morrisett, and Ryan Wisnesky. Trace-based verification of imperative programs with i/o. *Journal of Symbolic Computation*, 46(2):95, 2011.

[38] Per Martin-Löf and Giovanni Sambin. *Intuitionistic type theory*, volume 9. Bibliopolis Naples, 1984.

[39] Bengt Nordström, Kent Petersson, and Jan M Smith. *Programming in Martin-Löf's type theory*, volume 200. Oxford University Press, Oxford, 1990. Out of print.

[40] Ulf Norell. Dependently typed programming in agda. In *International School on Advanced Functional Programming*, pages 230–266. Springer, 2008.

[41] Cyrus Omar, Ian Voysey, Ravi Chugh, and Matthew A Hammer. Live functional programming with typed holes. *Proceedings of the ACM on Programming Languages*, 3(POPL):14, 2019.

[42] Giuseppe Peano. *Arithmetices principia: nova methodo exposita*. Fratres Bocca, 1889.

[43] Gordon Plotkin and Matija Pretnar. Handlers of algebraic effects. In *European Symposium on Programming*, pages 80–94. Springer, 2009.

[44] reddit user denito2. So i translated part of the 1st chapter of adam chlipala's book code from coq into c++ template metaprogramming... https://godbolt.org/z/bZTZrK, 2019.

[45] Ilya Sergey, Aleksandar Nanevski, and Anindya Banerjee. Mechanized verification of fine-grained concurrent programs. https://software.imdea.org/fcsl/, 2015.

[46] Gordon Stewart, Anindya Banerjee, and Aleksandar Nanevski. Dependent types for enforcement of information flow and erasure policies in heterogeneous data structures. In *Proceedings of the 15th Symposium on Principles and Practice of Declarative Programming*, pages 145–156. ACM, 2013.

[47] Nikhil Swamy, Joel Weinberger, Cole Schlesinger, Juan Chen, and Benjamin Livshits. Verifying higher-order programs with the dijkstra monad. In *ACM SIGPLAN Notices*, volume 48, pages 387–398. ACM, 2013.

[48] Marco Syfrig. Dependent types: Level up your types. https://eprints.hsr.ch/577/1/MarcoSyfrigDependentTypes_eprints.pdf, 2016.

[49] Carst Tankink, Herman Geuvers, James McKinna, and Freek Wiedijk. Proviola: A tool for proof re-animation. In *International Conference on Intelligent Computer Mathematics*, pages 440–454. Springer, 2010.

[50] Philip Wadler. Programming language foundations in agda. In *Brazilian Symposium on Formal Methods*, pages 56–73. Springer, 2018.

[51] The Agda wiki. http://www.cs.chalmers.se/~ulfn/Agda, 2018.

[52] Hongwei Xi. Programming with dependently typed data structures. 1999.

[53] Hongwei Xi. Applied type system: An approach to practical programming with theorem-proving. http://www.ats-lang.org/, 2017.

Towards an Artificial Intelligence Assistant for Software Engineers

Pedro M. F. Peixoto
Faculdade de Engenharia da Universidade do Porto, Portugal
up201802219@fe.up.pt

***Abstract*—Software Engineers are in a constant struggle to maintain themselves up to date with so much information scattered throughout the world and appearance of new knowledge every day. Numerous programming languages, tools and platforms exist and are actively used. Along with different design patterns and methodologies commonly used to solve known problems. Creating a constant growing demand, and tends to increase, for large knowledge bases. It is extremely difficult to contain all that knowledge without consultation. Since it agglomerates different technologies, programming languages, methodologies, methods and even user experiences. Given this problem, a description of possible features for an artificial intelligence assistant is presented. So, it can be publicly discussed, analyzed, adapted and used in future research in this area. Description of existing tools, technologies, limitations, functionalities and knowledge that can be integrated into an artificial intelligence assistant for software engineers is also presented. In conclusion identification of existing gaps and the advantages of having an artificial intelligence assistant that can communicate as a pair programmer is discussed, and future work proposed.**

***Index Terms*—Artificial Intelligence Assistant, Parallel Programming, Knowledge database, Pair Programming**

I. INTRODUCTION

The following subsections contain an introduction to human and machine interactions, the motivation that lead the research to take place and the contribution.

A. Human and machine

Human interaction through direct contact, written symbols and speech is a crucial aspect of human success. Contrary to other animals, it is believed that genetic code and spoken language is a result of evolution (or genetic modification). Other symbol systems were invented by humans such as written language, Arabic numerals, music notation and labanotation (notation system for human movement) [1]. Humans can create communication systems within their own species. Humans appear to distinguish themselves from most animals because of these abilities to express its ideas, emotions, sounds and even numeric values.

Machines are not living beings or entities. Interaction of people and machines is usually done with peripheral devices such as mouse and keyboard as well as sensors such as cameras and microphones. The events of the peripheral devices, frames of cameras and frequencies of microphones are then passed to a binary language that the machine is capable of processing. Arguably, machines are an extension of human intelligence, not individual self-aware beings. Uncapable of proposing philosophical questions to themselves such as "Je pense, donc je suis" [2].

In contrast, human beings usually do one task at a time and attempt to focus on that one given task to successfully complete it. Machines in the other hand, are not susceptible to distractions. Capable of executing multiple tasks at the same time. Parallel programming is a good example of multitasking. By dividing tasks among the existing processors.

With the public emergence of Text-to-Speech services, its use has been generalized [3], [4]. It can be used to read out loud small summaries or portions of text, to entire books. Allowing the illusion of a conversation or communication between human and machine to occur. For that to happen, huge amounts of knowledge are needed. Knowledge and easy access to it is a mankind dream that can be dated around two centuries B.C., with the construction of Alexandria Library [5]. According to literature it contained between 400k to 700k scrolls. That dream still endures. "World Brain" was the first known formal description of a true encyclopedia that could be accessed by everyone [6]. An idea published in 1938, written by H. G. Wells. An idea in a form of an article, or a science fiction genre. It was impossible to achieve such a feat at that time.

The problem today remains. A lot of virtual content is still difficult to be accessed and time consuming. For that reason, possible functionalities of an artificial intelligence assistant to help software engineers in their work is discussed here. Composed of theories based in existing literature and what functionalities should the artificial intelligence assistant have. How should it assist the individual, and what knowledge base is expected to have, so it can work efficiently. The objective of the assistant is to help the software engineer in his work.

B. Motivation

Worldwide libraries and online encyclopedias available to everyone are key arguments for proper social and scientific evolution and growth. So that this mankind dream can be achieved, an initial spark must be made. Artificial intelligence assistants must mature to take advantage of huge volumes of data, knowledge bases and achieve an improved artificial intelligence, maybe even cognitive intelligence.

Software engineering, as well as other areas, require lots of queries to be made to knowledge bases such as web, books, articles and other professionals. Feedback expectations in how to acquire this information has been increasing, especially with the significant impact of millennials generation [7]. This generation is more addicted to technology. More impatient, they desire answers quickly. Current solutions are not meeting the expectations.

Artificial intelligence assistants hold the promise of providing information and help software engineers. Knowledge is difficult to process, and some is enclosed within private companies and individuals. New knowledge must be collected and structured to answer this problem. A multi-source library should be available, but it does not exist.

Retrieving structured information about the queried subject is a challenge. Raw data needs to be correctly transformed into information that can later be recognized by the entity that queried it. Use of artificial intelligence and cognition intelligence has been proven reliable. Not only in simple voice queries made by the user, where the expectation is correct answers. But true pair programming feedback. An equivalent of querying another software engineer. There is no evidence in literature of a similar approach being implemented, researched or theorized.

C. Contribution

This paper reveals the major gaps, limitations and author design ideas for future development and implementation. It will hopefully raise constructive criticism and attract attention to the problem at hand. There is no evidence found in literature that could suggest the existence of a similar assistant, either research or tool. It is a contribution to future development of generic and professional virtual assistants in work places. In this case, software engineers.

In the remaining of the paper, the reader will find related work that is directly and indirectly connected to artificial intelligence assistants and software engineering. In chapter III the proposed functionalities for an artificial intelligence assistant is presented. Followed by discussion and conclusion chapters.

II. RELATED WORK

Artificial intelligence is already being used in software engineering in very discrete forms and places. This chapter is focused in describing the current state of the art in artificial intelligence for software engineering, as well as other useful tools and methodologies that can be integrated into assistants. The following Table 1 summarizes the related work presented in this section.

TABLE I
SUMMARY OF THE RESEARCH TOPICS

Relevant topics of AI Assistant for SE	Use of the topics and references		
	Applied in SE	As a virtual PP Assistant	Ref
Personal assistants	No	No	[8]-[12]
Tools	Yes		[13]-[16]
Data Structures			[17]-[21]
Education			[22]
Security			[23]
Quality and Testing			[24]-[28]
Tools			[29]-[36]

Please note that none of them is used as a virtual pair programmer assistant, to help software engineers, but most of them are used in software engineering.

A. Personal Assistants

Natural dialogue between machines and humans was first proposed in science fiction movies. In recent years it has become a reality. Virtual personal assistants can now communicate with humans to accomplish tasks such as open applications, execute certain commands, use search engines and even answer certain questions that are within the reach of its knowledge base. Multi-modal dialogue systems are also available [8]. These types of personal assistants allow combined user input to improve the quality of its services. Arguably, the most popular virtual personal assistants are Amazon's Alexa, Apple's Siri, Microsoft's Cortana, Google's Assistant and IBM's Watson assistant.

Multi-modal techniques include: gesture recognition, that are captured with cameras or with use of gloves; frames recognition from images or videos where context extraction is being explored; speech recognition improvements; text to speech, speech to text and conversational knowledge base. For deeper understanding and review of some specifications please consult the following references [8]–[10]. For an older example in how to execute commands and key bindings using voice you can consult one of the first applications ever made, GlovePIE [11].

Driving assistants and automobile assistants are also worth mentioning. Ranging from fuel consumption and driving advices, to self-driving cars and self-flight crafts such as Cora [12], [37], [38]. Assistants are taking an important role in vehicle optimization, with huge support from major brands and start-ups.

Big data computing and life sciences are also leveraging the use of cognitive computing [29]. Watson technology was used to discover relationships between biological entities. In life sciences large volumes of data need to be analyzed and understood. The results were promising, suggesting that Watson could accelerate identification of new drugs.

Appearance of artificial intelligence assistants, and desire of instantaneous feedback can also be traced from the rapid increase in technology and the millennial generation demands and expectations [7]. Easy access to technology has addicted the users even in their routine tasks. According to literature, there is a higher disinterest and shorter attention span [7].

B. Programming Tools

Personal assistants for generalized public for simple tasks are indeed in growth. In terms of assistants for specialized areas there is a lower rate. Software testing, product quality, modelling, code analyses and autocomplete are some of the tools that are still evolving in their own terms.

Autocomplete tools were game changing in software development and they still show a lot of promise. A great example of this kind of tools are Kite and Intellisense. Intellisense is part of Visual Studio and has been evolving for many

years. Kite is a python tool, that appears to have taken a small advantage over the competition [13]. Besides being an autocomplete tool, it also shows code use examples and samples of the functions you are using.

Fig. 1. Kite showing a code sample

The previous figure (Fig. 1) demonstrates the capabilities of having instant information while you are writing. The right-side panel presents detailed information about how to plot. Without ever needing to go search the web, guides, or books.

Software testing has grown to be a course in its own terms [39]. Over the years testing has proven to be a reliable tool to improve not only the quality of code, but also its longevity. Error detection and more importantly, earlier detection, using methodologies such as test driven development is definitely a method software engineers have grown accustomed to [40]. Tools and assistants in software testing also have their place in literature [41], [42]. Over the years these assistants have evolved and found its place into integrated software development environments [24]–[26]. Visual studio enterprise also contains live unit testing that allows unit tests to be generated and ran during code edition.

In graphical user interfaces, use of automation to test is an efficient way to recursively use test suites to detect code and behavior anomalies. It is also a growing area, which has significantly improved the life of testers. Since its mostly automated, the time consumption of error detection is decreasing, allowing the tester to do parallel work. The work found in literature verifies the interface events and triggers [27], [28]. Tools such as iMPAcT makes use of reverse engineering to crawl the application and determine which user interface test pattern should be applied [28]. The strategy is described in three phases that work in an iterative way: reverse engineering, pattern matching and testing. It is a fully automatic black box process. This type of applications is even capable of presenting a detailed report of the errors. This is a clear example of a tool that could become part of an assistant.

Live coding (or live programming) is improving over the years. Feedback received in certain technologies is almost instantaneous. Like for example changing XAML code or HTML code and visualizing the changes in the moment. Giving the feel of liveness in programming. An interesting perspective on its evolution can be found in literature [14]. Describing the different hierarchies when it comes to live coding, and the importance for software engineers to receive instantaneous feedback. Suggesting that with proper response times, coding can be done more efficiently.

Natural language artefacts were also discussed in literature [30]. In this article, it is proposed a solution to analyze unstructured natural language contained in the files. Such as commits, messages and comments. A plug-in was built for Eclipse integrated software development environment. Leaning on an interesting perspective in how interaction between human and machine could be done. With a pure focus on artifacts that are contained in the product, but not in the source code.

Programming assistants with the aim of **educating** and assist novice programmers can be found here [22]. In this example the objective was to help programmers learn parallel programming. This approach consists in training the solution to answer questions that the novice programmers may have, using natural language.

Natural language interfaces to databases allow users to query databases using natural language. This is an impressive achievement given the system needs to understand what context the query applies. It must keep track of the history of previous questions and answers, so it can make sense to the user. The article from literature presented a dialog interface to overcome this challenge [31].

C. A few Software Engineering Principles for Assistants

Unified modelling language has been crucial to object-oriented development. From database to activity diagrams. These diagrams can be generated during brain storms to facilitate development, marketing or simply consulting. A good combination of tools to leverage unified modelling language is OctoUML [43]. Supports collaborative software design and voice commands.

Database first approach is arguably the most popular approach when designing a database. Tools to design database diagrams that generate scripts to create the tables are also quite popular. Code First approach (also known as forward engineering) is less popular but has other advantages. Writing the classes (entities or object models) first and consequently generate the database can be used in new or existing databases. Facilitates the workflow by allowing the software engineer to write only the classes, and with commands generate the database, or migrations [15]. The opposite is also possible with current existing tools, reverse engineering of legacy databases to create an object model [16].

System and software quality are also a key component in every software engineer work. Proper quality evaluation is a laborious task. A standardized set of characteristics and objectives had to be set and used to achieve good quality assessment of all its content. Identifying what kind of behavior is expected, if the functionalities requested were implemented, determining what components are worth testing, evaluation of the software in terms of performance or even have a system to grade the individual components. For that reason, a common agreed consensus written and maintained by the international organization for standardization and the international electrotechnical commission, is present in literature [44]. A

strong example of a good knowledge base to help professionals evaluate properly theirs and third-party products.

There are other important principles in software engineering that should exist in an assistant such as software testing, graphical user interfaces patterns, architectural patterns, maintenance. But these were the most relevant.

D. How can data be stored and distributed

Collection of data from multiple sources and use of server-side computation and information providers to create big data has been and will continue to grow. Constantly growing knowledge databases can be leveraged to offer solutions to multiple tasks. Because this knowledge can be processed and converted into information.

Implementation of centralized data, web services and world-wide communication systems, allowed artificial intelligence to improve people's life by offering functionalities and information in their everyday routine. With the increase of knowledge database volumes, clustering machines and parallel programming has been found in literature as possible solutions for the lack of resources [17].

The other side of the coin is **decentralized data** using blockchain which gained popularity with the use of bitcoin [18]. Blockchain technology is a secure ledger of transactions among a network of computers. Companies like IBM and Microsoft Azure already provide business solutions. One of the main differences is that all data is distributed among different nodes. Contrary to centralized data where all data is stored in within a single provider. More information about security and decentralized data can be found throughout literature [19, 21].

Security can also be viewed independently of how the data is stored. Being a constant struggle to maintain devices secure, and services reliable, there is a need for strong knowledge bases to provide data and information about threats. A good example in literature for this area is precisely by creating a malware repository so they can be used by others [23]. Either professionals, researchers or generalized public. The provided reference explains how collected valuable forensic data, in this case about android applications, could be of interest to other parties.

E. Data Mining and Machine Learning Towards Cognitive Solutions

Formal theories in artificial intelligence and robotics are also present in literature [32, 33]. Underlining the importance of knowledge base to make proper **decisions** and **suggestions** [32]. Formal software development process was also discussed in terms of how can artificial intelligence improve modelling and proof in software development [33]. Taking advantage of data mining proofs and proof strategy languages. The volumes are so big, and numerous that techniques of machine learning, and data mining are now required to analyze the data [17], [45]. Evolution of cognitive solutions, and its aptitude to harness volumes of data lead to IBM Watson being used in big data [29]. Reaching to a conclusion that cognitive computing may be able to identify event reports from articles. As well as add efficiency to the research process.

Another successful example can be found in genetic and genomics research. Field that is strongly rooted in statistics [34]. Where the authors provided an overview of **machine learning** applications applied in their research field. Analysis of genome sequencing data sets.

Worldwide web is a very large source of information. Nowadays you can search a high variety of subjects but at the same time, it is difficult to acquire exactly what you desire, given the offer is so abundant. **Web crawling** is used to search and extract certain content from the pages. In literature, there are already solutions that can solve the problem of huge amount of data being collected. With the use of highly scalable applications [35]. Categorizing the information is also a solution to minimize unnecessary data to be collected [36].

F. Existing Gaps and limitations

Vehicles are being built in a way, so they can become drones, self-driven. At least for most of the routine trips. Programming may take a similar path, where certain routines tasks can be made, written and maintained by computers. Computers with enough capabilities that can be called Entities. No evidence was found about a machine or virtual assistant that could program properly without any kind of direct interference by a software engineer.

Autocomplete and code analysis are amazing tools. But these tools are not yet mature enough to be considered a replacement of another programmer. Another argument can be made then. There is also no evidence that an artificial intelligence assistant currently exists that could take the place of a programmer in pair programming approach.

Information is scattered among the internet, books, guides and professional experience. Searching through that data and transform it into information can be painful and time consuming. A race started by google to digitalize books, has transformed its search engine into one of the best in the world. This era is marked by big data. But its (real) use, is still gated. Construction of a knowledge database cannot be pointed out as an existing gap. But extraction tools, data mining tools, to extract data and transform it into information that can be provided to the software engineer is a valid argument.

Communication between machine and human is improving. Most progress is focused in personal assistants to query browsers or make appointments. Self-awareness, cognitive responses, memory through experience, ability to learn from the environment, improvisation and independence merged into a single artificial intelligence assistant cannot be found in literature.

Existing virtual assistants are using centralized data, within private companies. The code libraries are usable by programmers, but the knowledge base is contained and gated within the companies. There is no reference found about virtual assistants that take advantage of distributed data, blockchain. Where the community may be the main supporter and provider.

III. POSSIBLE FEATURES OF AN ARTIFICIAL INTELLIGENCE ASSISTANT

The author main goal is to pave out possibilities, for future work in this area. In no way the author of this paper is attempting to state that these are the only possible solutions. The author is merely expressing his intent and vision towards a possible working artificial intelligence assistant.

Pillars of software development such as planning, analysis, design, development, implementation, software testing, software quality and maintenance are present in different methodologies and processes used by professionals. Depending on the method of work used by the software engineers, the cycle and order may change. The assistant for software engineers was modelled based in those pillars. Necessary requirements and observations are also described throughout the next chapters.

A. Pair programming

In an ideal working place, every individual would have a co-worker with far more expertise than the software engineer. This would allow a higher level of confidence in software development. Unfortunately, it is quite common to have programmers working alone, or with co-workers with similar experience. The proposed approach of the artificial intelligence assistant for software engineers is equivalent to an extra professional in the team. An approximation of the idealized world brain, but as an element of the team [6]. An entity that can efficiently support the software engineer.

Teams and collective effort are a common practice among different areas. Technological evolution is no different and is a result of the combination of resources and knowledge. One fact that sometimes passes unnoticed is that technology itself is becoming an entity. One could argue that it is becoming an important element of the team, in some cases irreplaceable. Only difference is that it is an element to support others, and not to act on its own (for now).

Development environments are used to develop computer programs with interaction of a singular person or a collection of people. When performing these actions as a team, their knowledge is limited to their collective knowledge, capability of expression and communication and the time the team must collect new information (to learn something new). With an interactive virtual assistant, development teams could benefit from an extra element to improve their workflow.

To make artificial intelligence assistant a reliable support entity it cannot decrease the software engineer efficiency. Machine resources must be enough if both are sharing the same machine. Meaning the software engineer must program and the virtual assistant cannot negatively impact its development. Another valid argument is the assistant should be entirely apart from the programmer machine. Not only it will never negatively impact the programmer machine, but also be able to synchronously respond to multiple questions. Meaning the artificial intelligence assistant can respond to multiple queries, at the same time, from different programmers. Real time reading permissions from all the approved programmer machines could also open the possibility of prediction and warning messages. Considering that the entire team is working in the same project.

Collect data, controlling schedules, analyzing and predicting multiple machines is an achievement on its own. With visual studio live share extension, the code being written can be read, edited at the same time, and even some local resources are available and shared [46]. Allowing synchronized collaboration between different programmers in different machines, in real-time development. Meaning that an external machine dedicated to the artificial intelligence assistant can also use this type of technology.

B. Functionalities from the perspective of software development

Planning of the project dictates the success and effectiveness of the consequent stages. Usually a group of individuals are the ones who will decide and plan most of project. A brainstorm. In this stage is very difficult for the machine to interact or collect data from a conversation, files or frames. Even with the recent advancements of cognitive computation, cognitive processes and overall data collection. Understanding a conversation is also more complex than simple queries. That is why virtual personal assistants mainly respond to queries with links or answer to very direct questions.

To collect data, the virtual assistant requires sensors. On top of that, the artificial intelligence assistant also needs to comprehend the data, convert it from data to information. If the previous requirements are impossible, then a questionnaire should be considered. If one of the individuals that is present in the meeting can fill the questionnaire, then the artificial intelligence assistant can start making exclusion of parts, as well as improve possible answers to future queries. In a way, it needs contextualization, because natural language can be very deceiving. If it can create a questionnaire, to question the rest of the team, it will improve its ability to provide answers.

In terms of **analysis** and **design**, a strong working memory may take its most important role. With the project plan available to the artificial intelligence assistant, it can analyze and compare it with previous projects. A strong knowledge database and correct use of predictive modelling and machine learning techniques can provide valuable intel to the software engineer. Large amounts of memory and predict all possible outcomes are a few of the strengths of today's computing capabilities [29].

To properly analyze and design solutions, a reliable knowledge base must be available. Simple queries using a web search engine that provide links is not a reliable source of information. Neither is providing only answers to questions that have paired questions.

Development, **implementation** and **testing** are arguably the stages where artificial intelligence assistants can be most useful. Test-driven development has been shortening these stages, by finding errors earlier on. Still, it has not yet removed the need of testing, by other teams or individuals. To make sure the final products are as much bug free as possible. Predict every single possible outcome when writing a method will

depend on the experience of the programmer. Virtual assistants can aid in that regard by offering constructive comments about the methods, while they are being written. Independently if it is test driven development or not. An important observation to take note regarding software testing is time consumption to do the task. It is an essential and required chore but it's very repetitive.

Software engineering tasks are laborious, some of them extremely complex. But a fraction of them are routines. Grunt work, that is usually pushed to junior software engineers. Some of those tasks are: creation and use of template projects; scaffolding; database migrations; writing classes; writing small and simple methods with few lines of code; project updates. Some of these tasks could be made by the virtual assistant with the use of text or voice commands. However, teaching the artificial intelligence assistant enough commands so it can give the programmer the illusion of a pair programmer will be a challenge. It usually takes several years for a programmer to gain confidence and experience to execute certain tasks in a given technology, due to the steep learning curve, and vast knowledge required. Converting natural language or even syntax-based language to code will require structured knowledge databases, abstract knowledge base and respective data mining.

Other tasks that require the user attention cannot be made in parallel due to its complexity or importance. Current assistants can only offer constructive feedback or remind the user with messages and warnings. Assistants have been popular in this regard. Not only in reminding the users of its importance, but also by giving them suggestions and solutions. Similar to the virtual personal assistants who act as secretaries [8]–[10]. The proposition here, is to not only read and suggest changes during code writing, but also make certain changes in different branches of the original source code. Providing the option for the programmer to decide if he wants to use the code or not.

C. Customization

Most applications nowadays allow users to customize its personal experience, for that reason, **customization** is also a requirement. Allowing the user to configure permissions, functionalities, personal messages, personal commands and selection of limitations. Reader should consider adding functionalities in an iterative and incremental method [47]. Every time a new functionality is incremented, respective options should be added. Collective options for teams and communities should also be present and available.

D. Limitations

Artificial Intelligence assistant should only have access to the environments that the user gave permissions. Permissions can vary between users: read; write; suggest; observe; report; interact. The permissions to the virtual assistant should be controlled too. **Security** breaches, **responsibility**, **accountability** are all strong reasons, that require attention to properly function within a team of engineers. Therefore, it is proposed that the virtual assistant is contained in its own machine, with limited permissions, and should not have direct access to write in other user machines. This can be achieved with branches of the original source code, ask for permission before doing any kind of action that should be trackable and accountable and have proof of concept and decisions made. When someone asks the virtual assistant to do something, the accountability should fall in the software engineer who order it. A reliable system that proves it was asked by that professional is also required.

From the perspective of **knowledge value** and security. Ideally, the resources allocated to the artificial intelligence assistant should be protected from prying eyes. Over time, the data collected from the virtual assistant will increase in value. The more specifications and customization the company or individual adds to the assistant the more efficient it becomes. Knowledge is power. Losing or illegally acquiring that data could reveal potential sensitive data about the company. Meaning that there are limitations in the way data is transferred between nodes (if it is decentralized), or when it reaches the servers (big data). It must be secure, and safely stored. This is truly a controversial problem. It is very difficult to provide security, in an unsecure network. Meticulous research will have to be done in this topic to achieve a balanced solution.

E. Knowledge Base and Databases

Data management passes through collection of articles, books, media, guides, templates, code excerpts, snippets, common mistakes, errors and personalized messages. These are some of the content that should be contained within the knowledge database. An example of some of the content can be found here [48]. Enabling the ability to query a knowledge database such as w3schools, and efficiently answer those queries with either code examples or explanations would be a great motivational start [36]. This would be a great improvement over Kite, since it gives a voice interface and becomes much more than just a autocomplete tool [13]. It becomes a source of knowledge.

Collecting personal and collective data is a controversial subject. A centralized knowledge database and provider could hold most of the data. But has shown in related work, decentralized data also has its advantages [21]. A decentralized approach for this model seems more appropriate given the fact that knowledge is of extreme importance and should be protected at all costs. As well as it would be in the hands of the community, available to all.

In terms of how to collect the data for the knowledge base. A common aspect of human communication, natural language, is somehow being underrated in existing applications. Collection of data directly from the user should be made in conventional formats. Natural language is also considered in the design of the system, it will increase its value. And decrease the barriers between human and machine, since it gives the user an alternative to provide data.

Personal and collective metrics should also be collected when a tool is used, such as the assistant. These metrics and statistics can be of great use for collective teams. As well

as open source communities that could share their data to improve the services.

A **ranking system** also needs to be considered during the design. Every time a query is made, the user should be able to contribute with a metric. Dictating if the result obtained was relevant or not. This score dictating if it is relevant or not will improve the quality of the system. Committing a new block to the chain stating the new score of the relation between query and answer. Statistics about queries that are constantly receiving negative feedback can also be analyzed through this system and improve accordingly. On top of that, when no suggestion is made, the user can also suggest an answer and added it to the chain.

The knowledge database systems should be designed with specific intents. It is important to collect data to be reused but is also important to give power to the user, so they can decide what to do with it. The most important goal is to decipher what is of interest to collect from a scientific growth perspective, and not from a commercial point of view. The following paragraph describes some of the sources of information that may be used.

The assistant can acquire, and save data from sites, using web crawling tools (sites contain a lot of information that can be used) [36]. Collection of personal and collective data with the use of metrics and surveys. Allow data to be deleted if requested. Allow custom made documents such as voice commands, snippets and messages to be stored in a collective or personal perspective. Possibility of connection to different types of repositories that contain articles, books, guides, surveys or other future relevant source of information.

Knowledge base will increase over time. An important feature is functionalities to allow the knowledge base to increase during user usage. Reader should also consider maintenance strategies to apply over time. It is very difficult to predict what exactly will happen, and how will it evolve. Beware of legal issues when it comes to how the assistant acquire that data.

F. Data Mining? The anwser Towards Cognitive Intelligence?

Different types of data will have to be processed. From videos, images, books, articles, guides, sites, code snippets, excerpts of code and project templates. Portions of data will be structured, but the high majority will be unstructured data. A strategy to query, analyze and acquire intelligence from the knowledge base needs to be further researched and designed [17], [45]. Cognitive intelligence will also have an important role in supplying information [29].

Predictive analytics may also be of great value when certain data is not yet in existence. When working with new technologies, there will be no data to obtain. In Fig. 2 the data will flow to the artificial intelligence component of the assistant. However, that may not be enough. Cognitive intelligence is required to evolve the assistant to another level. Abstract problem solving, learn from experience and capability to resolve unpredictable events is a category that will require further research. Cognitive intelligence is usually attached to humans, not machines. Improvements in this subject will allow the assistant to integrate more functionalities. Such as: formulation of new ideas or solutions that are not present in the knowledge bases; creation of new knowledge; and automated programming.

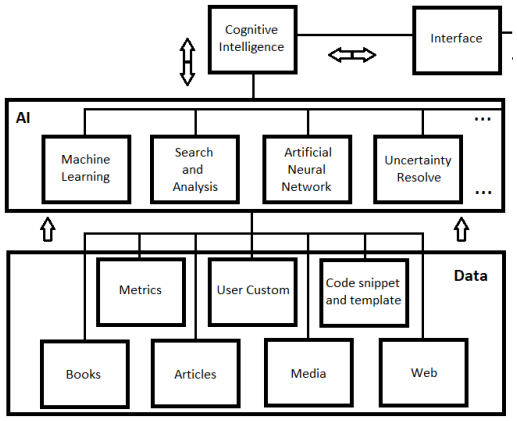


Fig. 2. Summarized structure of the components

Achieving this mile stone will take a lot of effort. The current computer architecture may not even allow such feature to ever exist. Human brain and machine are not compatible. So, the initial processes towards cognitive intelligence, will have to continue with artificial intelligence equivalents. Since the development and implementation of efficient techniques to acquire information will be a long incremental and iterative process. Time will pass by, until the next category is required and achieved, making it capable of structuring it it-self, as an entity.

G. Legal and Lawfull Acts and Perspectives

This type of technology can be extremely evasive. The data collected can be, but not limited to: screen captures; remote access; live share of local resources; voice recordings; phone calls; email messages; browser history; working hours; resting hours; break times; application usage; social media; calendar and schedule; work efficiency. This type of data is required to improve the quality of services provided to the users. Depending in how this data is stored and related to the user, it may be considered personal. Meaning it may be under the general data protection regulation or similar laws. From an engineering perspective, the system needs to be designed so that the data collected is approved, or denied, by the user. If it is approved, the user needs to have the option to permanently delete it at any given time. Or collect data anonymously, without any relation to the users [49].

IV. FINAL TOUGHTS ON SOME OF THE PRESENTED CONTENT

A theoretical research describing possible functionalities for an artificial intelligence assistant to help software engineers was presented. Revealing a few gaps in the existing technologies and literature. Following paragraphs discuss some important points, that the reader should take note.

Assistants can collect massive amounts of data about the users. In terms of security, encryption of all personal data is a must have. In some cases, is required by law. To decrease the probability of permanently losing the data it is advised to create backups. Another interesting alternative is with use of encrypted blockchains [21]. Where the information is distributed between nodes instead of centralization (big data).

Distributed data and use of shared data among the nodes could also be a significant step in the right direction. Not only the data would be less likely to be lost, but it would also improve the quality of the services. Since there is a portion of data that is common to all users. Shared, maintained and distributed by the community.

Integrating commands (text and voice) to execute these tasks could improve software development efficiency. Commands can generate almost instantaneously a task, process or method. More importantly, those events can be done in parallel. The struggle however, may not be proving that it will improve the engineer's workload but make it accountable after so much automation is added.

Improving cognitive intelligence for virtual assistants is crucial. Every time the programmer reaches a point where he may lack information to continue his job, the artificial intelligence assistant should be able to detect that lack of knowledge. Eliminating the requirement of the engineer to know the commands, or code syntax. The ability to solve problems and reason about the software engineer decisions is also very important to align with the ideal of pair programming with a virtual assistant.

V. CONCLUSION AND FUTURE WORKS

The proposed functionalities have the potential to improve the quality and efficiency of software engineer's work and maybe even their life quality. This is a summarized version of the research done given the limitation of content. That provides information of some functionalities that the assistant should have to help software engineers. This is a theoretical research with the intent of challenging the scientific community to an even bolder research, an engineer research. So that virtual assistants can continue to improve. There are a lot of barriers that still need to be broken.

In conclusion here are some final thoughts. Parallel programming and machine clustering are described in literature as an efficient solution to high consumption processes and high-volume knowledge bases. Suggesting that it is a possible solution to be used in creating an entity capable of helping the software engineer.

A virtual assistant can be a candidate to pair programming and potential element of the team, it can support the programmer without ever overstepping the programmer and will not impact its performance in a negative way. Its knowledge base should be reliable and available. Distributed data, in most cases, present more advantages than centralized data.

Artificial intelligence assistants are far from being considered entities. Detailing the functionalities was the first step. Further research must be done to pass this model from a theoretical research to a working prototype. An artificial intelligence assistant as a pair programmer has never been discussed or idealized in literature. Given the rise of assistants in other areas, this may be a reliable alternative to the conventional methods in a near future. Combining virtual assistants, reliable knowledge bases, and artificial intelligence techniques this type of tools can be the next step of evolution in software engineering.

REFERENCES

[1] D. Premack, "Is Language the Key to Human Intelligence?," Science, vol. 303, no. 5656. pp. 318–320, 2004.

[2] R. Descartes, "Discours de la méthode: pour bien conduire sa raison et chercher la vérité dans les sciences," Französisch Deutsch, 1637.

[3] W. Hamza, R. Bakis, E. Eide, M. Picheny, and J. Pitrelli, "The IBM expressive speech synthesis system," Conf. Int. Speech Commun. Assoc., pp. 1099–1108, 2004.

[4] J. F. Pitrelli, R. Bakis, E. M. Eide, R. Fernandez, W. Hamza, and M. A. Picheny, "The IBM expressive text-to-speech synthesis system for american english," IEEE Trans. Audio, Speech Lang. Process., vol. 14, no. 4, pp. 1099–1108, Jul. 2006.

[5] H. Phillips, "The Great Library of Alexandria?," Libr. Philos. Pract., 2010.

[6] S. Degoutin and G. Wagon, "World Brain," Societes, 2015.

[7] E. B. Aguirre and S. D. F. Jr, "Lived Stories of Mid-Career Teachers: Their Struggles with Millennial Learners in the Philippines," vol. 8, no. 1, pp. 39–50, 2018.

[8] V. Kepuska and G. Bohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)," 2018 IEEE 8th Annu. Comput. Commun. Work. Conf. CCWC 2018, vol. 2018–Janua, no. c, pp. 99–103, 2018.

[9] M. B. Hoy, "Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants," Med. Ref. Serv. Q., vol. 37, no. 1, pp. 81–88, Jan. 2018.

[10] G. López, L. Quesada, and L. A. Guerrero, "Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces," Springer, Cham, 2018, pp. 241–250.

[11] C. Kenner, "GlovePIE," 2010. [Online]. Available: https://sites.google.com/site/carlkenner/glovepie. [Accessed: 09-Jan-2019].

[12] V. C. Magana and M. Munoz-Organero, "Artemisa: A Personal Driving Assistant for Fuel Saving," IEEE Trans. Mob. Comput., vol. 15, no. 10, pp. 2437–2451, Oct. 2016.

[13] "Kite - AI-Powered Python Copilot." [Online]. Available: https://kite.com/. [Accessed: 06-Jan-2019].

[14] S. L. Tanimoto, "A perspective on the evolution of live programming," in 2013 1st International Workshop on Live Programming, LIVE 2013 - Proceedings, 2013, pp. 31–34.

[15] L. Naylor, "Using Entity Framework Code First with an Existing Database," in ASP.NET MVC with Entity Framework and CSS, Berkeley, CA: Apress, 2016, pp. 407–426.

[16] H. Schwichtenberg, "Reverse Engineering of Existing Databases (Database First Development)," in Modern Data Access with Entity Framework Core, Berkeley, CA: Apress, 2018, pp. 37–59.

[17] D. E. O'Leary, "Artificial intelligence and big data," IEEE Intell. Syst., vol. 28, no. 2, pp. 96–99, Mar. 2013.

[18] N. Satoshi and S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic cash system," Bitcoin, 2008.

[19] H. Orman, "Blockchain: The emperors new PKI?," IEEE Internet Comput., vol. 22, no. 2, pp. 23–28, Mar. 2018.

[20] G. Zyskind, O. Nathan, and A. "Sandy" Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015, 2015, pp. 180–184.

[21] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts," in Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016, 2016, pp. 839–858.

[22] S. Memeti and S. Pllana, "PAPA: A parallel programming assistant powered by IBM Watson cognitive computing technology," J. Comput. Sci., vol. 26, pp. 275–284, 2018.

[23] M. Bierma, E. Gustafson, J. Erickson, D. Fritz, and Y. R. Choe, "Andlantis: Large-scale Android Dynamic Analysis," in In Proceedings of the Third Workshop on Mobile Security Technologies (MoST) 2014, 2014.

[24] N. Tillmann and J. de Halleux, "Pex–White Box Test Generation for .NET," Springer, Berlin, Heidelberg, 2008, pp. 134–153.

[25] N. Tillmann, J. de Halleux, and T. Xie, "Transferring an automated test generation tool to practice: From Pex to Fakes and Code Digger," in Proceedings of the 29th ACM/IEEE international conference on Automated software engineering - ASE '14, 2014, pp. 385–396.

[26] G. Fraser and A. Arcuri, "A Large-Scale Evaluation of Automated Unit Test Generation Using EvoSuite," ACM Trans. Softw. Eng. Methodol., vol. 24, no. 2, pp. 1–42, Dec. 2014.

[27] I. C. Morgado and A. C. R. Paiva, "The iMPAcT tool: Testing UI patterns on mobile applications," in Proceedings - 2015 30th IEEE/ACM International Conference on Automated Software Engineering, ASE 2015, 2016, pp. 876–881.

[28] I. C. Morgado and A. C. R. Paiva, "Mobile GUI testing," Softw. Qual. J., vol. 26, no. 4, pp. 1553–1570, Dec. 2018.

[29] Y. Chen, J. Elenee Argentinis, and G. Weber, "IBM Watson: How Cognitive Computing Can Be Applied to Big Data Challenges in Life Sciences Research," Clin. Ther., vol. 38, no. 4, pp. 688–701, Apr. 2016.

[30] R. Witte, B. Sateli, N. Khamis, and J. Rilling, "Intelligent software development environments: Integrating natural language processing with the eclipse platform," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2011.

[31] A. Mittal, J. Sen, D. Saha, and K. Sankaranarayanan, "An Ontology based Dialog Interface to Database," in Proceedings of the 2018 International Conference on Management of Data - SIGMOD '18, 2018, pp. 1749–1752.

[32] S. J. Rosenschein, "Formal theories of knowledge in AI and robotics," New Gener. Comput., vol. 3, no. 4, pp. 345–357, Dec. 1985.

[33] A. Bundy, D. Hutter, C. B. Jones, and J. Strother Moore, "AI meets Formal Software Development."

[34] M. W. Libbrecht and W. S. Noble, "Machine learning applications in genetics and genomics," Nature Reviews Genetics, vol. 16, no. 6. Nature Publishing Group, pp. 321–332, 07-Jun-2015.

[35] G. C. Deka, "NoSQL Web Crawler Application," in Advances in Computers, vol. 109, Elsevier, 2018, pp. 77–100.

[36] J. Pruthi and Monika, "Implementation of category-wise focused web crawler," in Advances in Intelligent Systems and Computing, 2018, vol. 654, pp. 565–574.

[37] M. Bojarski et al., "End to End Learning for Self-Driving Cars," Apr. 2016.

[38] Kitty Hawk, "Cora." [Online]. Available: https://cora.aero/. [Accessed: 09-Jan-2019].

[39] S. H. Edwards, S. H., Edwards, and S. H., "Using software testing to move students from trial-and-error to reflection-in-action," in Proceedings of the 35th SIGCSE technical symposium on Computer science education - SIGCSE '04, 2004, vol. 36, no. 1, p. 26.

[40] T. Bhat and N. Nagappan, "Evaluating the efficacy of test-driven development," in Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering - ISESE '06, 2006, p. 356.

[41] W. Schulte, "Pex–An Intelligent Assistant for Rigorous Developer Testing," in 12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007), 2007, pp. 161–161.

[42] G. Fraser and A. Arcuri, "EvoSuite," in Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering - SIGSOFT/FSE '11, 2011, p. 416.

[43] R. Jolak, B. Vesin, and M. R. V Chaudron, "Using voice commands for uml modelling support on interactive whiteboards: Insights & experiences," in CIbSE 2017 - XX Ibero-American Conference on Software Engineering, 2017.

[44] "ISO/IEC 25010:2011 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models." [Online]. Available: https://www.iso.org/standard/35733.html. [Accessed: 15-Dec-2018].

[45] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," Int. J. Inf. Manage., vol. 35, no. 2, pp. 137–144, Apr. 2015.

[46] Microsoft, "Visual Studio Live Share." [Online]. Available: https://visualstudio.microsoft.com/services/live-share/. [Accessed: 10-Jan-2019].

[47] C. Larman and Craig, Agile and iterative development: a manager's guide. Addison-Wesley, 2004.

[48] "W3Schools Online Web Tutorials." [Online]. Available: https://www.w3schools.com/. [Accessed: 08-Dec-2018].

[49] C. Laybats and J. Davies, "GDPR," Bus. Inf. Rev., 2018.

Toward a Soccer Server extension for automated learning of robotic soccer strategies

Miguel Abreu
FEUP (DEI) - University of Porto
Rua Dr. Roberto Frias, 4200-465 Porto – Portugal
up201809115@fe.up.pt

***Abstract*—A solution for an optimal robotic soccer strategy is yet to be found. Multiple agents interacting in an environment with continuous state and action spaces is a recipe for machine learning lethargy. Fortunately, on the one hand, due to the increasing hardware performance and new algorithms, the area of reinforcement learning is growing considerably. On the other hand, simulating the environment for strategy purposes is not following this trend. Several simulators are available, including those used in major soccer competitions (e.g. RoboCup). However, no option combines a good repository of teams with a simple command set that abstracts low-level actions. To clarify this problem, we surveyed the most promising simulators and proposed a preliminary extension for the well-established Soccer Server. The objective was to simplify the process of learning strategy-related behaviors through automated optimization algorithms. The results have shown a clear advantage in using the extension to improve the agent's performance. These results were confirmed through an ablation study which emphasized the most important components of the proposed extension. This work contributes to the development of future strategies related with RoboCup or other soccer competitions. Despite the good results, there is space for improvement in computational efficiency and behavior diversity.**

***Index Terms*—simulation, soccer, robotic, multi agent, strategy**

I. Introduction

The high-level dynamics of simulated soccer is extremely complex. The task of creating a good strategy for a team is challenging enough for experienced developers, but it can certainly overwhelm newcomers. Fortunately, the field of reinforcement learning has grown considerably in the last years, allowing agents to learn in complex environments without prior knowledge.

However, while targeting high-level tasks such as team strategy, it is hard to find a simulator which combines simplicity with a repository of teams with respectable results. The concept of simplicity can be deconstructed into two main ideas. First, the simulator must not lose computation resources on tasks which are not relevant for the learning process. Second, it must provide abstract commands so that the learning algorithm does not need to worry about basic actions (e.g. dribbling, rotating, etc.). There are several simulators related with major soccer competitions [1]–[4], but none is optimized for high-level behaviors.

Having a repository with high-quality opponents is also an important metric since they can be leveraged at different parts of the project development. During learning, the initially naive team can acquire years of human expertise by devising ways of surpassing state-of-the-art algorithms. Even if the learning from scratch strategy is adopted (as in AlphaGo Zero [5]), the repository will become advantageous in the testing stage of the project. It can be used as a benchmark to assess the performance against a non-evolving opponent.

The main objective of this paper is to present a simulator that simplifies the process of learning strategy-related behaviors through automated optimization algorithms. As initially mentioned, the simulator should abstract low-level actions, be computationally efficient and provide a stable benchmark infrastructure. To achieve this goal, several current solutions will be reviewed to build an improved alternative. For testing and assessment, the outcome of the previous step will be used to learn the notions of ball possession, field progression and enemy avoidance.

A. Summary of Contributions

- The most relevant sports simulators are compared in-depth, targeting their adequacy for automated learning of high-level robotic soccer strategies (section II);
- The advantages and disadvantages of each approach are taken into consideration when devising an extension for the Soccer Server simulator. The resulting abstraction layer is described in section III-A and the methods used to test our solution are presented in section III-B.;
- The abstraction layer is tested using the Proximal Policy Optimization algorithm to train a single player against agents from two RoboCup 2018 teams (section IV-A). An ablation study is performed to assess the importance of each new command (section IV-B);
- The results are discussed in section V;
- Finally, some conclusions and future work plans are established (section VI).

II. Related Work

Advances in machine learning and the constant improvement of hardware performance have led to an increasing interest in simulation tools, in an attempt to leverage new techniques in complex scenarios. The same is true when referring to simulated soccer. The Robot World Cup Initiative (RoboCup) is a very well known international robotics competition which gathers teams from many countries annually since

1997 [6]. It was considered the first competition to promote research and education using soccer games [7].

A. Soccer Server

First released in 1995, the Soccer Server has been the official simulator for the competition, having incorporated new functionalities along the years [8]. It enables autonomous agents written in different programming languages to play soccer against each other in real-time, in a mixed cooperative-competitive 2D environment. These agents are implemented as single clients which connect to the server via UDP/IP sockets [1]. Each client specifies actions for the corresponding player using predefined control-commands and receives its sensor readings. It may also communicate with other clients using `say` and `hear` commands.

In Fig. 1 is shown the architecture of the soccer server, according to Noda et al. [1]. The message board module is responsible for all the communications to and from the server. The referee module implements all the soccer rules and the field simulator module is the physics engine.

It is possible to display a virtual field using a program called Soccer Monitor, which is part of the Soccer Server. Multiple monitors may be active at the same time using the X window system (shown in Fig. 1) to communicate with the server.

B. Simspark

RoboCup is also composed of a 3D Soccer Simulation League (3DSSL), for which SimSpark is the official simulator since the league's first competition in 2004 [2], [9]. With the added realism comes the inevitable inherent complexity of the physics engine, which is the Open Dynamics Engine (ODE), as seen in Fig. 2. The simulation engine allows agents to connect through an IP network or directly as plugins, which simplifies debugging and reduces overhead.

Although SimSpark is an interesting simulator to learn realistic motor skills, the deterioration of performance makes it harder for machine learning algorithms to learn high-level behaviors, such as strategy and positioning. It would require a way of abstracting motor skills (e.g. walking, passing, shooting) with predefined algorithms. This raises two major issues. First, the algorithms for these behaviors are in constant evolution and, by definition, they cannot be considered

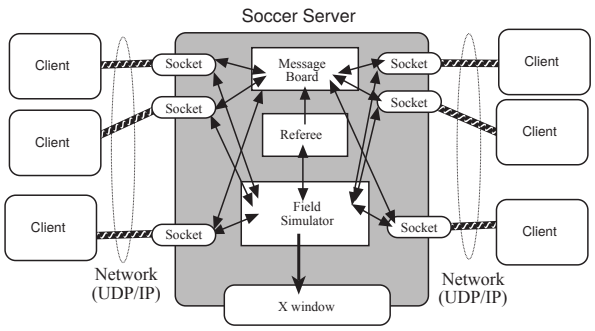


Fig. 1. Architecture of the Soccer Server [1]

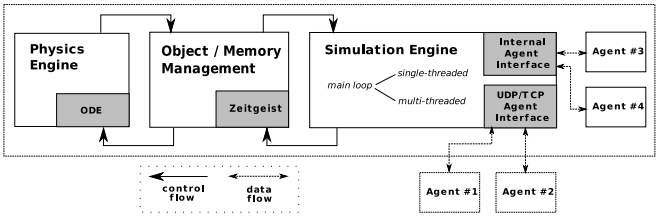


Fig. 2. Main components of Simspark (excluding graphics) [9]

optimal. Second, the computation of such interactions with the environment would cause unnecessary overhead.

C. The Tao of Soccer

In 2001, Zhang released the first version of The Tao of Soccer (TOS), a 2D soccer simulator which allows human players to battle against teams of AI programs [10], [11]. Like the two previously introduced simulators, TOS has as open architecture, allowing the user to write a client program in various languages, as long as it supports network communication through UDP sockets.

Since the program admits human intervention, low level actions are actively simplified by the server, promoting confrontations on a strategic level. This feature is very appealing, considering the objective of this work. However, apart from the bundled AI opponents, there is only one featured team in the simulator's website repository.

D. RoSoS

RoSoS is an open-source soccer simulator created in 2015 for virtual robots tournaments [12], [13]. It was designed to motivate young students to take an active part in robotics, and improve the quality of education. With this in mind, it implements a straightforward method to control the players, without compromising the 2D simulation's realism. It's coded in Java and it can be easily modified to implement new rules, physics algorithms and other game aspects.

Martins et al. have organized a small competition in Brazil, in July 2015 with 8 competing groups. Each group shared their code, which is accessible through a common repository. Additionally, the project is bundled with some rudimentary example teams.

Regarding locomotion of each player, the user may opt for the omnidirectional model or differential-drive. The former is a better option to abstract the walking process, when learning a strategy, since it does not depend on the previous value of the player's direction. Another feature of this simulator is the absence of the kick action. The player must push the ball by planning strategic collisions to steer it in the correct direction. Although the list of actions is very small, it may require more complex control patterns.

E. In-depth Comparison

1) Environment and Actions: The relevant implementation details of the previous simulators were gathered in Table I. Regarding the environment and possible actions, the presented

simulators cover a good set of options. The Soccer Server allows many low-level "play on" actions (i.e. actions that are allowed when the game is not stopped by the referee). Each player must be able to coordinate its movement (translation and rotation) while kicking or tackling (depending on the situation) without loosing sight of what is happening around it (which involves turning the neck because the environment is only partially observable).

Although realistic, this approach is very complex considering that only the strategy is being learned. However, the Soccer Server is composed of a trainer module (also called offline coach), which is not allowed in official games but can be very useful for automated learning [14]. The trainer solves the agents' visual limitations by providing noise-free data about all players and ball.

SimSpark has similar sensory information regarding other movable objects, albeit adapted to three dimensions. The body information is highly detailed, including accelerometer, gyroscope, joint perceptors, etc. The available actions abide by the same abstraction level, only allowing the agent to control the speed of each robot's joint.

The Tao of Soccer combines the translation and rotation movements into the `drive` command, and eliminates the `catch` and `turn_neck` commands (because each agent has a 360 visual field range). In older versions, close dribbling was implemented automatically by the server as the player moved, effectively simplifying low level coordination. When several players disputed the ball, the simulator would randomly choose one of them to keep possession. The dribbling control in recent versions is more complex and thus, it will not be covered.

In RoSoS, the agent may set a constant translation and rotation speed, which is kept until it sends a `stop` command. There is no coach and there are less rules, when comparing with all the previous simulators. Another unique feature is the simpler set of sensors offered by the default robot, including a compass and ball sensor, and 4 distance sensors located around the player to detect movable objects and field walls. Although apparently easier to control, each agent cannot distinguish teammates from opponents.

2) Testing Framework: The previous sections reviewed the agent's perceptions about the environment and the available actions. Now, the focus is set on the simulator's performance, the possibility of extending each server and respective modules, and the repository of qualified teams.

All the previously proposed projects are well documented, easy to modify or extend and open source. The complexity of the Open Dynamics Engine makes SimSpark server the slowest one, obtaining the relative performance grade "C" in Table I. The simpler physics engine of the Soccer Server yields the performance grade "B". The best grade was given to TOS, mainly due to the client/server protocol, which has great impact on the overall testing framework. It was "largely simplified" and produces "less network traffic" and "higher performance", when compared to the Soccer Server, according to Zhang [10]. Due to the absence of performance comparisons in literature, RoSoS was not graded.

Regarding repositories of qualified teams, the Soccer Server and SimSpark take the first prize, with 394 binaries for the Robocup's 2D league and 182 binaries for the 3D league, organized by year [15]. The other simulators have very small repositories, ranging from a few qualified teams to none.

F. Conclusion

We considered the Soccer Server as the most adequate simulator, mainly due to two reasons: its large, high-quality repository of teams, and its fast and realistic physics engine. However, since it lacks a simple interface, we propose an extended version, inspired in the features of the other presented simulators. By virtue of the Soccer Server documentation's dimension, its rules and mathematical models will not be listed. However, a certain degree of familiarity is recommended before delving into the next section (see [8], [14]).

III. Methods

In this section are described the extensions that will be performed to the Soccer Server to allow strategy learning algorithms to bypass low-level actions.

A. Abstraction Layer

The abstraction layer was created to extend the abilities of the Soccer Server without losing compatibility with teams from the official competition repository. Its integration in the extended architecture is depicted in Fig. 3. The server recognizes "high-level teams" (i.e. teams that play on a strategic level) if their name starts with the initialism "HLT". In this context, if an agent calls a `dash` command, it will be using the abstraction layer.

In the automated learning stage, the existing trainer eliminates the need for vision related commands, namely, `turn neck`, `pay attention to` and `point to`. The last two commands were used to lock the focus of a player on another player, or on a specific point. At each control cycle, the trainer acquires absolute positions and velocities of all moving objects (i.e. the ball and all players from both teams). Then, it sends the data, via the `say` command, to every HLT agent.

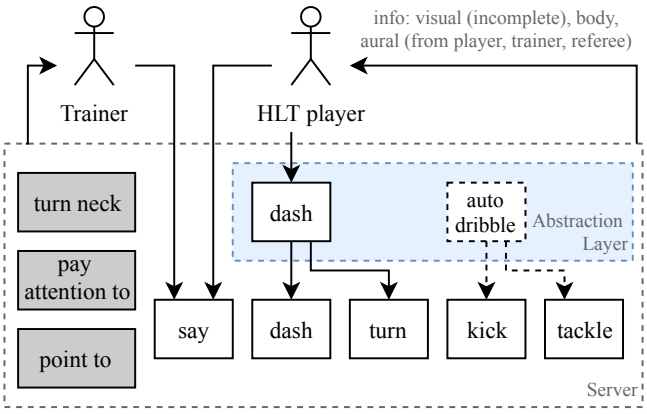


Fig. 3. Extended Soccer Server control architecture

TABLE I
SIMULATORS IMPLEMENTATION DETAILS

	Soccer Server	**SimSpark**	**TOS**	**RoSoS**
"Play on" actions	catch (goalkeeper), dash, kick, pay attention to, point to, say, tackle, turn, turn neck	joint speed	drive, kick, message	move, rotate, stop
Sensory Information	visual (incomplete), aural[a], body	visual (incomplete), aural[a], body	visual (360°), aural, referee	distance, compass, ball
Team Repository	394	182	1	8
Server Performance	B (2D)	C (3D)	A (2D)	- (2D)
Project	C++ (open source)	C++ (open source)	Java (open source)	Java (open source)

[a] in these cases, referee messages are received as aural data

1) Dashing: The original `dash` command had two parameters – `direction` and `power`. The abstract version also has the same parameters but the latter is translated to speed instead of acceleration. This process takes into consideration the player's stamina level, as before. Initially, the player is rotated in the desired direction, and then it is moved forward. Combining direction with the computed speed yields the polar vector of velocity. Comparing with the older version, the movement differences are as follows:

- Since the player is first rotated, the direction is removed from the physics movement model. Therefore, there is no loss of power related with dashing sideways or backwards, since it is always considered a forward movement;
- The player's previous velocity does not affect its current behavior, which simplifies the optimization algorithm's task. This is the advantage of converting the power to speed instead of acceleration. The underlying physics model is not broken, since this approach is equivalent to creating an acceleration $\vec{a}$ that compensates the observed velocity $\vec{v}$, before applying the desired speed s and direction d:

$$\vec{a} = (s, \angle d) - \vec{v} \ . \tag{1}$$

- The stamina still decreases, as desired, since it dissuades aimless running, but the extra energy consumption associated with dashing backwards is no longer applied.

2) Dribbling: Dribbling is performed automatically. This behavior includes regaining ball possession without the need for a `tackle` command. First, the algorithm checks which player is closest to the ball. Then, if that player is from a HL team, the algorithm will verify if it has the ball's possession. This process is divided into the following steps:

1) The distance between player and dribble point is measured, as in Fig. 4. Distance d is obtained from the multiplication of the player's current velocity vector $\vec{v}$ by a constant k_1. In practice, this means that the dribble point gets farther away as $\|\vec{v}\|$ increases.
2) The possession area is enclosed by a circle of radius r, with its center on the dribble point, where r is equal to a constant k_2.

If the ball is inside the possession area, it will be attracted to its center (the dribble point). The idea is to simulate the concept of possession for different case scenarios. In a real situation, when the player is running fast, it must kick the ball with greater power, thus loosing the tight control it had. The farther the ball, the easier it is to be intercepted by the opponent. Note that the dribbling algorithm stops attracting the ball if a non-HL agent is closer to it.

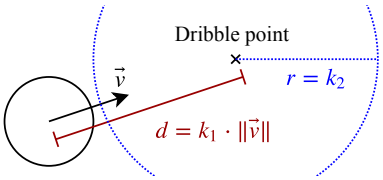


Fig. 4. Computing the dribble point's distance by multiplying the player's current velocity vector length by a constant k. The possession area is enclosed by a circle of radius r, with its center on the dribble point.

After some testing to find natural looking behaviors, the values of k_1 and k_2 were fixed at 15 and $2m$, respectively. The high value of k_1 is explained by the way the player's velocity is calculated at each cycle. The server computes the dribble point before moving every object. Therefore, the player's velocity is obtained from the previous step, multiplied by a decay factor, according to the Soccer Server's movement models [8]. To avoid losing momentum when dribbling at close range, the collision between ball and HLT players was disabled.

This concept is better than grabbing the ball, for two main reasons. First, the game physics is not broken by teleporting the ball to a different location at every time step, while loosing its original speed and ignoring obstacles. Second, the player is dissuaded from running towards the goal at full speed while employing some absurd evasive maneuvers to avoid being tackled. The attraction idea reminds the player to be careful, since fast reactions may lead to loosing control of the ball.

When the ball is kicked or tackled by an outfield player, or punched by the goalkeeper, the automatic dribbling algorithm is deactivated for 5 cycles. This prevents the dribbling attraction from interfering with the ball's trajectory, and allows non-HL teams to steal the ball with less effort. The algorithm is also deactivated while the ball is caught by a goalkeeper. From the methods presented above, it is possible to conclude that the player's direction has a reduced influence on its actions, apart from defining the attraction point when dribbling the ball.

B. Testing methods

1) Proximal Policy Optimization: Proximal Policy Optimization (PPO) is a reinforcement learning algorithm introduced by Schulman et al. in [16]. This method was chosen due to its performance and ease of tuning. It is known for the objective it optimizes, which can be translated by the following equation:

$$L(\theta) = \hat{\mathbb{E}}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t], \quad where \;\; r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}(a_t|s_t)}}, \tag{2}$$

where $\hat{A}_t$ is an estimator of the advantage function at timestep t. The clip function clips the probability ratio $r_t(\theta)$ in the interval given by $[1-\epsilon, 1+\epsilon]$. The parallel implementation of PPO from the OpenAI repository [17] was used. It alternates between sampling data from multiple parallel sources, and performing several epochs of stochastic gradient ascent on the sampled data, to optimize the objective function.

2) Benchmark Framework: The benchmark framework is depicted in Fig. 5. The PPO algorithm obtains data from multiple parallel games simultaneously. Each game is played between an instance of the HL team and a repository team. Playing against different opponents in parallel games is a technique used to mitigate overfitting issues. For performance reasons, each HL team instance is composed of a single thread. The server produces observations and sends them to the trainer via UDP. These are then delivered to the PPO algorithm, which yields abstract actions for each player. The single-threaded approach reduces network traffic by eliminating the need for the trainer to communicate through the `say` command.

The above described method has some practical limitations. The strategy learned with the extended simulator cannot be used directly in official RoboCup competitions, due to the abstraction layer. However, the purpose of the extension is not to train a fully-fledged team. Nonetheless, the resulting strategies may be integrated into exiting teams.

3) State space: Instead of a distributed approach, a centralized decision core was adopted to commands the entire HL team. This choice was based on the class of behaviors that is being learned. Multi-agent communication is not a part of this paper's scope, and thus the strategy will be coordinated exclusively by an offline coach (trainer).

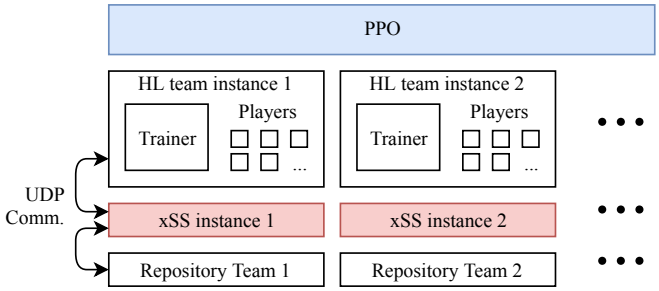


Fig. 5. Benchmark framework

The state space contains only two-dimensional continuous features, as seen in Table II. The first two features correspond to the position and velocity of the ball, which is part of the environment. Considering the set of players $P = \{p_1, p_2, ..., p_n\}$, there are $2 \cdot n$ additional features, indicating the position and velocity for each element of P.

4) Action space: The action space only comprises two parameters per player, indicating the desired power vector coordinates. This vector is used for the abstract `dash` command.

5) Episode Setup: Initially, the game mode is set to `before_kick_off`. The agent is placed on a random position, on the left half of the field. Its x and y coordinate ranges are $[-20, -10]$ and $[-20, 20]$, respectively. The ball is placed $2m$ in front of the agent. Then, some time is awarded to the opposing team to ensure that every player's position was reset to the initial values. Finally, the game mode is changed to `play_on` and the episode begins.

6) Reward: To provide a reference point for the agent, a target was created inside the opponent's goal area, at $0.5m$ from the field's edge. More specifically, its x and y coordinates are $53m$ and $0m$, respectively. The reward is given by $\Delta d(b, t)$, which represents the distance variation between ball and target, hereinafter referred to as ball's distance to target (BDTT).

7) Terminating conditions: The episodes end if at least one of two terminating conditions is met. The first one checks if the ball is out of play, under the official RoboCup rules. The second condition checks if the maximum number of time steps has passed. This limit was set to 200 steps (20 seconds).

IV. Results

The abstraction layer was tested using the benchmark framework depicted in Fig. 5. The optimization algorithm was used to train a dribbling experiment against two 2018 RoboCup teams, using different actions to assess their relevance in learning the desired behaviors.

A. Dribbling

In this experiment, a HLT agent must dash, as fast as possible, in the direction of the opponent team's goal. The objective is to keep the ball's possession by leveraging the auto dribble feature. The opponent may steal the ball by kicking or tackling. In case the opponent is not HL, these actions may be performed when the opponent is closer to the ball (since the dribbling feature is disabled) or by taking advantage of the 5 cycle dribble deactivation, explained in section III-A2.

TABLE II
State features names and data types

State Feature	Description
Ball position	2D point
Ball velocity	2D vector
Player p position ($p \in P$)	2D point
Player p velocity ($p \in P$)	2D vector

The objective of this experiment is to test the extended simulator's capabilities, namely, the abstract `dash` command and the `auto dribble` feature. It will serve as a baseline to compare with the next section, in which an ablation study will be presented. To build this baseline, a HLT agent will play in 3 scenarios:

- Hillstone - the agent will face 5 outfield players and the goalkeeper from the Hillstone 2018 team, classified in 11th place in RoboCup 2018 Soccer Simulation 2D League. Training will be performed with two instances running in parallel, using the same repository team (see Fig. 5).
- Helios - the conditions are the same, but the team is Helios 2018, classified in 1st place in the same competition;
- Mixed - the agent plays against one instance of each team presented above.

The fields depicted in Fig. 6 enclose representations of the areas occupied by each player for different experiments. For each scenario, the presented data was obtained after the optimization was concluded, for a total of 50 consecutive episodes. Although the sampled values were continuous, they were discretized to improve the plot's presentation. The optimized agent is represented by red circles and the opponents by triangles, which were painted in different shades of gray. As an exception, the opponent playing in the most advanced position is depicted in green to emphasize the importance of its role.

1) Hillstone: Fig. 6**a** corresponds to the behaviors developed while facing the Hillstone 2018 team. The agent has learned to attack from the flanks and it never goes through the center. In 50 episodes, the opponent team has successfully scored 3 times. Table III shows the results obtained in this experiment, where the first line corresponds to the current scenario. After the optimization was concluded, each agent was tested on 1000 episodes, except for the Mixed scenario which was tested twice against each repository team. In the current scenario, on average, the ball's distance to target (BDTT) was reduced by $41.8m$ with a standard deviation (SD) of $16.3m$. On 3.0% of the attempts, that distance was increased. The two extreme BDTT variations were -$55.6m$ and $49.1m$. The total traveled distance by the agent was, on average, $57.8m$ with a SD of $5.8m$.

2) Helios: The Helios 2018 team has a different strategy, as seen in Fig. 6**b**. The goalkeeper plays on a more advanced position, as well as the outfield players. The agent also learned to attack from the flanks, but, on average, the BDTT increased $3.0m$ with a SD of $23.7m$. Its variation ranged between -$30.7m$ and $49.1m$, being negative on 39.2% of the attempts. The total traveled distance was, on average, $34.2m$ with a SD of $7.2m$.

3) Mixed: In this scenario the agent was trained against both repository teams simultaneously, in parallel sessions. The post-optimization analysis was conducted independently for each opponent, yielding the visual representations shown in Fig. 6**c** and 6**d**. In both cases, the agent learned to avoid the field's center. In comparison with previous scenarios, the average BDTT variation increased $1.2m$ against Hillstone and decreased $15.4m$ against Helios. To avoid repetition in this analysis, the remaining results for the mixed scenario can be seen in Table III.

B. Ablation Study

This study aims to evaluate the abstraction layer's importance in the learned bahaviors. The Mixed scenario experiment presented in the previous section was repeated without certain abstract features. This scenario was chosen due to the opponent's heterogeneity and the obtained results.

1) Original `dash`: First, the abstract dash command was deactivated. This means that the agent must directly use the server's dash command. Although the agent is not able to turn, the `auto dribble` feature is not affected. It uses the player's velocity vector to obtain the dribble point's angle, instead of its orientation.

The resulting patterns are shown in Fig. 6**e** and 6**f**. The data analysis can be seen in Table IV. Against Hillstone, the average BDTT increased 15.5% ($6.4m$) in relation to the original dribbling experiment. Against Helios, the relative variation was larger, standing at 39.5% ($4.9m$). The ratio between average traveled distance and absolute average BDTT increased from 1.38 to 1.52, and from 2.94 to 4.17, against Hillstone and Helios, respectively. Finally, the number of times the ball ended up farther from the opponent's goal also increased for both cases.

2) Dribble: In this experiment, the original Mixed scenario was reproduced without the `auto dribble` feature performed by the server. To compensate for this loss on the agent's side, an ad hoc manual dribbling algorithm was implemented. A comparison between both approaches can be seen in Fig. 7. For each option, three consecutive time steps are represented. The maximum dribbling distance d_{pb_max}, for the automated method, corresponds to the player's maximum velocity magnitude, as explained in section III-A2. The ad hoc algorithm kicks the ball if it is close enough. The kick power is equivalent to the agent's current dash power divided

TABLE III
DRIBBLING EXPERIMENT RESULTS FOR EACH SCENARIO

Scenario		**Avg. $\Delta d(b,t)$ ($\pm$1 SD)**	**Avg. traveled dist. ($\pm$1 SD)**	**% Positive $\Delta d(b,t)$**
Hillstone		-41.8 ($\pm$16.3) m	57.8 ($\pm$5.8) m	3.0%
Helios		3.0 ($\pm$23.7) m	34.2 ($\pm$7.2) m	39.2%
Mixed	(Hill.)	-40.6 ($\pm$20.2) m	55.9 ($\pm$5.6) m	5.5%
	(Heli.)	-12.4 ($\pm$16.6) m	36.5 ($\pm$6.7) m	11.1%

TABLE IV
ABLATION STUDY RESULTS (-ABSTRACT DASH)

Scenario		**Avg. $\Delta d(b,t)$ ($\pm$1 SD)**	**Avg. traveled dist. ($\pm$1 SD)**	**% Positive $\Delta d(b,t)$**
Mixed	(Hill.)	-34.3 ($\pm$19.6) m	52.1 ($\pm$8.1) m	6.2%
	(Heli.)	-7.5 ($\pm$14.6) m	31.3 ($\pm$7.7) m	15.7%

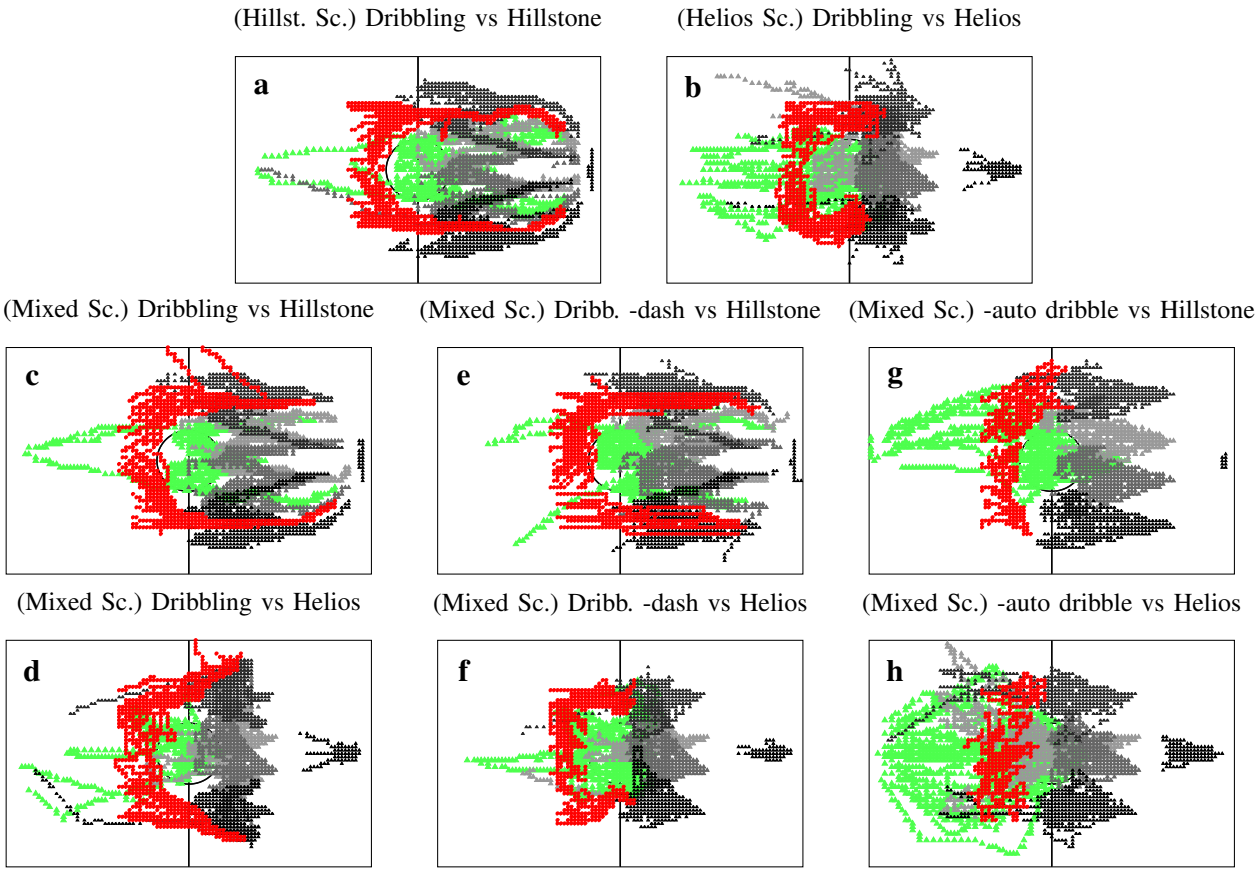


Fig. 6. Area of the field occupied by multiple agents, in 50 consecutive episodes, after the optimization was concluded. The HLT player (red circles) is trying to reach the opponents goal without losing possession of the ball. Hillstone players are represented by gray/green triangles. The data was discretized after collecting the samples to improve the scatter plot's presentation.

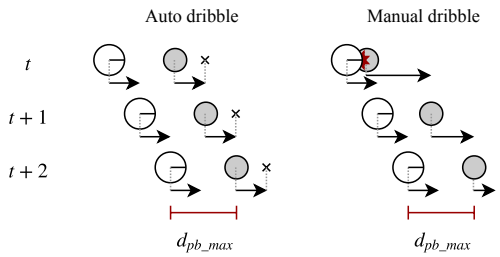


Fig. 7. Comparison between the auto dribble feature and an ad hoc manual dribbling algorithm, for three consecutive time steps. On both situations, the player (white circle) is moving at maximum speed. The ball (gray circle) is attracted toward the dribble point (× mark) when the auto dribble feature is enabled. Otherwise, it must be periodically kicked. The maximum distance between player and ball is denoted as d_{pb_max}.

by a constant k_3. To preserve the original d_{pb_max}, after an empirical analysis, k_3 was set to 2.7.

The results are shown in Table V and Fig. 6**g** and 6**h**. In both cases, the average BDTT became positive, in comparison with the baseline. Against Hillstone, the percentage of times the ball ended up farther from the opponent's goal rose from 3.0% to 31.0%. Against Helios, this percentage went from 11.1% to 97.1%.

TABLE V
ABLATION STUDY RESULTS (-AUTO DRIBBLE)

Scenario		**Avg. $\Delta d(b,t)$ ($\pm$1 SD)**	**Avg. traveled dist. ($\pm$1 SD)**	**% Positive $\Delta d(b,t)$**
Mixed	(Hill.)	6.7 ($\pm$22.1) m	13.1 ($\pm$5.5) m	31.0%
	(Heli.)	38.1 ($\pm$12.0) m	12.1 ($\pm$3.1) m	97.1%

V. DISCUSSION

A. Dribbling

As expected, the RoboCup 2018 Soccer Simulation 2D League champion was harder to beat than the 11th place finisher. It displayed a more aggressive style of play, closing down the space with nimble moves. Although this experiment aimed the creation of a baseline for the ablation study, some interesting results have emerged. Training against Helios and Hillstone simultaneously yielded better results than expected. Comparing with the Hillstone scenario, there was a slight performance reduction, although Fig. 6**a** and 6**c** show similar patterns. However, comparing with the Helios scenario, there was a noticeable improvement in distance and ball possession, as clearly depicted in Fig. 6**b** and 6**d**. This shows that the

learned policy, not only generalized for both teams, but also created more efficient patterns.

The agent learned to avoid the opponent, specially when playing against Helios. It was able to perceive direct confrontation as a possible source of danger. When facing a more aggressive opponent, the distance between players was reduced, and the agent consistently turned around for a brief period of time to avoid tackles. Another interesting outcome was the relation between this distance and the agent's velocity. Slowing down was another technique it used to avoid being tackled. Ultimately, the agent learned to adapt the dribble point's distance to favor field progression or evasiveness.

B. Ablation study

1) Original `dash`*:* Removing the abstract `dash` from the original Mixed scenario did not result is drastic performance losses. However, the results reflect the initial advantage given by the high-level command, to which the orientation is irrelevant. When using the original `dash`, going sideways or backwards has a negative effect on the agent's velocity, thus complicating the control task. Against Helios, the difference was more notorious. The ratio between the average traveled distance and the absolute average BDTT increased because the agent spent more time avoiding the opponent than progressing in the field.

2) Dribble: The auto dribble feature plays a very important role in learning high-level behaviors, since it gives the player additional control over the ball. This experiment has shown very different results for both opponent teams. Against Hillstone, the agent managed to reduce the distance to goal in most episodes, despite the slow progression. However, when facing the RoboCup champion, it performed very poorly, losing the ball almost every time.

VI. Conclusion and future work

The proposed Soccer Server extension is a step toward automated strategy learning. It provides agents with a solid lower level layer, which can be used as a base to build abstract notions, such as evasiveness, ball possession, field/game progression, etc. The results should not be interpreted as quality comparisons between learning agent and static repository teams, for several reasons. First, the abstraction layer gives a clear advantage to the learning agent. Second, the repository teams were not deployed using eleven players, which is their normal working state. Third, they are programmed to conserve stamina to last for long periods of time, while the learning agent only has to worry about the lenght of a small episode.

Accordingly, the results should be interpreted as benchmarks, to compare the efficiency of multiple configurations and the progression made by different optimization algorithms. This is possible due to the open access availability of static RoboCup teams binaries [15], and to the compatibility of the proposed abstraction layer.

The results have shown a clear advantage in using the extension to learn high-level behaviors. The agent was able to focus on a proper strategy instead of wasting time with irrelevant simulation particularities. It learned to keep possession by evading the opponent and using speed to its advantage. The behaviors were generalized to cope with the differences between two very different RoboCup teams. One of them was Helios, the RoboCup 2018 Soccer Simulation 2D League champion, and the other was Hillstone, classified in 11th place.

The ablation study has confirmed this positive perspective by emphasizing the importance of certain abstraction layer commands. In the future, the extension can evolve in several directions. Its architecture is ready to receive additional layers to further abstract its actions or simply increase their diversity.

Although the underlying Soccer Server allows the deactivation of certain features to increase the computational speed (such as the referee), there is plenty of work to do in this area. However, it is not easy to remove features or simplify the simulation models without affecting the compatibility with repository teams.

References

[1] I. Noda, H. Matsubara, K. Hiraki, and I. Frank, "Soccer server: a tool for research on multiagent systems," *Applied Artificial Intelligence*, vol. 12, no. 2-3, pp. 233–250, 1998.

[2] Y. Xu and H. Vatankhah, "Simspark: An open source robot simulator developed by the robocup community," in *Robot Soccer World Cup*. Springer, 2013, pp. 632–639.

[3] M. Hausknecht, P. Mupparaju, S. Subramanian, S. Kalyanakrishnan, and P. Stone, "Half field offense: An environment for multiagent learning and ad hoc teamwork," in *AAMAS Adaptive Learning Agents (ALA) Workshop*, Singapore, May 2016.

[4] The MagmaOffenburg RoboCup 3D Simulation Team. magmachallenge: Benchmark tool for robocup 3d soccer simulation. [Online]. Available: https://github.com/magmaOffenburg/magmaChallenge

[5] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[6] I. Noda, ji Suzuki, H. Matsubara, M. Asada, and H. Kitano, "RoboCup-97: The First Robot World Cup Soccer Games and Conferences," Tech. Rep., 1998.

[7] RoboCup Federation. A brief history of robocup. [Online]. Available: https://www.robocup.org/a_brief_history_of_robocup

[8] H. Akiyama. The robocup soccer simulator users manual. [Online]. Available: https://rcsoccersim.github.io/manual/

[9] J. Boedecker and M. Asada, "Simspark–concepts and application in the robocup 3d soccer simulation league," *Autonomous Robots*, vol. 174, p. 181, 2008.

[10] Y. Zhang. The tao of soccer. [Online]. Available: http://soccer.sourceforge.net/soccer/soccer.html

[11] A. Mackworth, Y. Zhang, M. Crowley, and Shane. The tao of soccer. [Online]. Available: https://sourceforge.net/projects/soccer/

[12] F. N. Martins, I. S. Gomes, and C. R. Santos, "RoSoS - A free and open-source robot soccer simulator for educational robotics," in *Communications in Computer and Information Science*, 2016.

[13] F. N. Martins, I. S. Gomes, R. Ferreira, and J. P. Vilas. Robot soccer simulator (rosos). [Online]. Available: http://ivanseidel.github.io/Robot-Soccer-Simulator/

[14] M. Chen, K. Dorer, E. Foroughi, F. Heintz, Z. Huang, S. Kapetanakis, K. Kostiadis, J. Kummeneje, J. Murray, I. Noda, O. Obst, P. Riley, T. Steffens, Y. Wang, and X. Yin, "Users manual: Robocup soccer server manual for soccer server version 7.07 and later," Tech. Rep., 2003.

[15] S. Glaser. robocup.info archive. [Online]. Available: https://archive.robocup.info/Soccer/Simulation/2D/binaries/RoboCup/2018/

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," pp. 1–12, 2017.

[17] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov, "Openai baselines," https://github.com/openai/baselines, 2017.

Evaluation of a low-cost multithreading approach solution for an embedded system based on Arduino with pseudo-threads

Juliana Paula Félix
Instituto de Informática
Universidade Federal de Goiás
Goiânia, Brazil
julianapaulafelix@inf.ufg.br

Ênio Prates Vasconcelos Filho
Cister Research Centre
Instituto Superior de Engenharia do Porto
Porto, Portugal
Instituto Federal de Goiás
Goiânia, Brazil
enpvf@isep.ipp.pt

Flávio Henrique Teles Vieira
Escola de Engenharia Elétrica, Mecânica e de Computação
Universidade Federal de Goiás
Goiânia, Brazil
flavio_vieira@ufg.br

***Abstract*—Although projects using Arduino boards are becoming more and more common due to their simplicity, low cost, and a variety of applications, Arduino boards consist of a simple processor that does not allow the execution of threads. This paper presents a study and evaluation of multithreading approaches on a single Arduino board. We present a group of existing software approaches for dealing with concurrent actions on Arduino. Among the solutions presented, we propose a case study using timed interrupts due to their simplicity. Although the case study provided requires dealing with many actions concurrently, including external actions, timed interrupts showed to be a robust solution to the problem. Furthermore, the evaluated approach presented great potential for being applied and implemented commercially at low cost.**

***Index Terms*—Embedded Systems, Multithreading, Arduino, Timed Interrupts**

I. Introduction

ARDUINO is an open-source electronic platform intended for anyone interested in creating interactive objects or environments [1]. Most Arduino boards consist of an Atmel 8-bit AVR microcontroller, varying its amount of flash memory, pins and features, which can be expanded by using shields. Shields are Arduino-compatible boards that can be plugged into the customarily supplied Arduino pin headers in order to provide other features, such as the addition of sensors, Ethernet, Global Positioning Systems (GPS), Liquid Crystal Displays (LCDs), and so much more.

Because of its simplicity and low cost, Arduino has been used for a wide range of applications, from simple projects such as blinking a set of LEDs, to more sophisticated ones, like controlling a robot arm [2] or a 3D printer [3]. Its applications can also be extended to commercial purposes [3–6]. Many universities are also using this device to introduce their students in the roles of programming, prototype and hardware development [7–9].

The Arduino is a very simple processor and has no operating system, which means it can run only one task at a time. In other words, it does not support threads. A thread of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler, which is typically a part of the operating system [10]. The implementation of threads and processes differs between operating systems. In most of the cases, however, a thread is a component of a process. Multiple threads can exist within one process, running concurrently and sharing resources, such as memory, while different processes do not share these resources. In particular, a thread of a process shares its executable code and the values of its variables at any given time.

Standard Arduino API provides a programming model based on a basic structure compounded by two functions: `setup` and `loop`. The `setup` function runs only once and is followed by the `loop` function which repeats indefinitely. Thus, `setup` is usually used to prepare the program environment, while `loop` plays a role as the main program, performing a set of actions. We highlight that this programming model is a single thread model.

Although this programming model can be straightforward and easy to be programmed, it might present challenges since some actions such as IO (input/output) and state switching can conflict with each other. For instance, a simple blinking LED, usually implemented by using the function `delay`, holds the processor for the time the LED is kept on or off, while a card reader, which depends on external actions, requires the program to wait for data input so that another action can be followed. In this scenario, if the system forces the LED to blink on a regular time basis, it risks not to wait enough time to read a card and lose a card reading trial. On the other hand, if the system waits a considerable amount of time trying to ensure recognizing a card reading trial, the LED blinking might be delayed or happen asynchronously.

This hypothetical scenario is certainly not the hardest one to be solved. However, in a more complicated situation, engineers can end up opting for using more than one Arduino board to distribute tasks and avoid conflicts. Moreover, this is not just a waste of processing ability, but it also increases project cost. Therefore, many programmers have been working on providing solutions for multithreading on microcontrollers. In this paper, we provide a study on available techniques for running concurrent threads effectively on a single Arduino

board, and we analyze a case study of a situation where it occurs, applying one of the techniques described.

This paper is organized as follows: In Section II, we discuss techniques allowing an Arduino to run more than one thread at a time, Section III presents our case study and the solution we provided for it. In Section IV, we discuss the case study and efficiency of timed solutions and, finally, in Section V we present our concluding remarks.

II. Multithreading Approaches for the Arduino Platform

Due to its simplicity, Arduino does not support real Threads (parallel tasks). However, there are many scenarios which systems have to deal concurrently with synchronous tasks, such as loops for showing status or blocking procedures such as user inputs. In this sense, many solutions can be found in the literature, but most of them are based on pseudo-threads. This section provides a general description of the most common approaches and their respective details.

A. Interrupts

An interrupt is a signal that tells the processor to immediately stop what it is doing and handle some high priority processing [11]. Once all code attached to an interrupt is executed, the processor goes back to whatever task it was initially doing before the signal happened.

If one wants to ensure that a program always catches the press of a button, it would be very tricky to write a program to do anything else, since the program would need to constantly check the status of the pin attached to a button. If any other task has to be performed, which consumes time, the risk of missing a button press increases significantly.

By using interrupts, the system can react quickly and efficiently to important events that cannot be easily anticipated in software, such as monitoring user input. Moreover, it frees up the processor and allows it to do other tasks while waiting for an event to occur. Examples of tasks that might be benefited if interrupts are used include reading an I2C device [1], reading a rotary encoder [2], sending or receiving wireless data and, of course, monitoring user input.

On Arduino, this is provided by the function `interrupts()`, which allows certain priority tasks to happen in the background, while the microcontroller can get some other work done while not missing a user input, for example. This function is enabled by default, and works along with the function `attachInterrupt`, which takes three parameters [11].

The first parameter to `attachInterrupt` is an interrupt number in which we specify the actual digital pin that will be monitored. For example, if one wants to monitor pin 2, they should use `digitalPintToInterrupt(2)` as the first parameter to `attachInterrupt`.

The second parameter, usually referred to as an ISR (Interrupt Service Routine), is the function to be called when the interrupt occurs. This function must take no parameters, and they should return nothing. Generally, an ISR should be as short and fast as possible, and there should be no `delay()` call, so as to not interfere with other routines that the Arduino must execute. If a sketch[3] uses multiple ISRs, only one can run at a time. Other interrupts can be performed after the current one finishes, and their order or execution depends on the priority each one of them have.

The third parameter defines when the interrupt should be triggered. Arduino documentation [11] says there are four constants predefined as valid values: LOW to trigger the interrupt whenever the pin is low, CHANGE to trigger the interrupt whenever the pin changes value, RISING to trigger when the pin goes from low to high, and FALLING for when the pin goes from high to low.

Typically, global variables are used to pass data between an ISR and the main program and are usually declared as volatile in order to make sure they are updated correctly. Another essential point about interrupts on Arduino is that their number is limited, depending on the Arduino board. On UNO, for instance, there are only two digital pins that can be used for interrupts (pins 2 and 3). On Mega 2560, six pins can be used for interrupts (2, 3, 18, 19, 20, 21) and this number can get higher on other boards, such as DUE and 101, where all digital pins can be used, the latter restricting only the mode it can operate.

B. Timed interrupts

A timed interrupt, as the name suggests, is an interruption that is triggered when a specified time interval has been reached, similar to an alarm clock that rings when the time previously set comes. In a timed interruption, one can set a timer to trigger an interruption at precisely timed intervals. When an interval is reached, an alert can be emitted, a different part of code can be run, or a pin output can be changed, for example.

Just like external interrupts, timed interrupts run asynchronously or independently from the main program. Rather than running a loop or repeatedly calling `millis()`, one can let a timer do that work for them while the code does other things. For instance, if one wants to build an application that changes the status of a LED every 5s while it constantly does other things, they can set up the interrupt and turn on the timer, allowing the LED to blink perfectly on time, regardless of what else is being performed in the main program.

The AVR ATmega168 and ATmega328, which are used by Arduino UNO, Duemilanove, Mini and any of Sparkfun's Pro series, have three timers: Timer0, Timer1, and Timer2. The AVR ATmega1280 and ATmega2560 (found in the Arduino Mega variants) have three additional timers: Timer3, Timer4, and Timer5. Timer0 and Timer2 are 8-bit timers, meaning its

[1] I2C stands for "Inter Integrated Circuit" and an I2C device has a bidirectional two-wired serial bus which is used to transport the data between integrated circuits.

[2] An electro-mechanical device that converts the angular position or motion of a shaft or axle to analog or digital output signals.

[3] A sketch is the name that Arduino uses for a program. It is the unit of code that is uploaded to and run on an Arduino board.

count can record a maximum value of 255. Timer1, Timer3, Timer4, and Timer5 are 16-bit timers, with a maximum counter value of 65535. The details on how to use them can be found at ATmega328 [12] and ATmega2560 [13] datasheets, respectively.

C. Arduino Threads

ArduinoThreads is a library developed by Seidel [14] for managing the periodic execution of multiple tasks. The library promises to simplify programs that need to perform multiple periodic tasks, providing a way to let the program to run "pseudo-background" tasks. The user defines a Thread object for each of those tasks, then lets the library manage their scheduled execution.

There are, basically, three classes included in this Library: `Thread`, `ThreadController`, and `StaticThreadController` (both controllers inherit from `Thread`). `Thread` class is the basic class, which contains methods to set and run callbacks, check whether the thread should be run, and also create a unique `ThreadID` on the instantiation. `ThreadController` is responsible for holding multiple thread objects, also called as "a group of Threads", and it is used to perform `run` of every thread only when needed. `StaticThreadController` is a slightly faster and smaller version of `ThreadController`. It works in a way similar to `ThreadController`, but once constructed it cannot add or remove threads to run.

The heart of this approach is the `ThreadController`, which has a `run` method. Each thread is run sequentially, thus if `ThreadController.run()` method is called from the main `loop`, the program will run identically to a conventional `setup`/`loop` Arduino program. Therefore, programmers must create a timed interruption to call the `run` method in order to make threads compete with the `loop` function, but not interfere with each other.

ArduinoThreads' project is more of a threading emulation which organizes the programmer's code in an object-oriented programming way. A single timed interrupt calling a function can do the same task without any new thing to learn, but it can keep the code cleaner if the programmer has some code building preference or if the code grows too much.

D. AVR-OS Library

AVR-OS [15], developed by Cris Moos as an open source project, provides a very basic run-time that enables a program to deal with multiple threads on Arduino Uno, Mega and Mega 2560. Although its name has OS on it, which can be thought of as an operating system, it is actually a library, and therefore there is no need to replace the Arduino bootloader, being fully compatible to regular Arduino sketches.

As main characteristics, AVR-OS uses pre-emptive multitasking to switch tasks, and each task has its own stack that is restored when a task is resumed. It implements a simple thread scheduler based on an AVR timer, in order to provide ticks to switch between tasks.

E. Protothreads

Protothreads are a programming abstraction that provides a conditional blocking-wait statement intended to simplify event-driven programming for memory constrained embedded systems [16]. Protothreads can be seen as a combination of threads and events, having inherited the blocking-wait semantics from the former and the stacklessness and low memory overhead from the latter, using as low as two bytes per protothread.

While protothreads were originally created for memory-constrained embedded systems, it has also been proven to be useful as a general purpose library. It has been used in the Contiki operating system [17], and by many different third-party embedded developers [16]. Examples include a MPEG decoding module for Internet TV-boxes, wireless sensors, and embedded devices collecting data from charge-coupled devices.

Protothreads provide linear code execution for event-driven systems. It is highly portable, the library is implemented 100% in C and uses no architecture specific assembly code. It can be used with or without an underlying operating system to provide blocking event-handlers [18], providing a sequential flow of control without complex state machines or full multi-threading. Finally, it is available under a BSD-like open source license and can be downloaded at Adam Dunkels' website [18].

F. RTuinOS

RTuinOS is an event based Real-Time Operating System (RTOS) for the Arduino environment, created by Peter Vranken and it is available currently on [19], since moved from [20].

As mentioned earlier, a traditional Arduino sketch has two entry points: the function `setup`, which is the place to put the initialization code required to run the sketch, and function `loop`, which is periodically called. The frequency of looping is not deterministic but depends on the execution time of the code inside the loop.

Using RTuinOS, the two mentioned functions continue to exist and continue to have the same meaning. However, as part of the code initialization in `setup`, one may define a number of tasks having individual properties. The most relevant property of a task is a C code function, which becomes the so-called task function. Once entering the traditional Arduino `loop`, all of these task functions are executed in parallel to one another, as well as parallel to the repeated execution of function `loop`. We say that the function `loop` becomes the idle task of the RTOS.

A characteristic of RTuinOS is that the behavior of a task is not entirely predetermined at compile time. RTuinOS supports regular, time-controlled tasks as well as purely event controlled ones. Tasks can be preemptive or behave cooperatively. Task scheduling can be done using time slices and a round-robin pattern. Moreover, many of these modes can be mixed.

A task is not per se regular, its implementing code decides what happens, and this can be decided based on the context

or the situation. To achieve this flexibility, RTuinOS has an event controlled scheduler, where typical RTOS use cases are supported by providing according events, e.g. absolute-point-in-time-reached. If the task's code decides to always wait for the same absolute-point-in-time-reached event, then it becomes a regular task. However, in a situation-dependent scenario, the same task could decide to wait for an application sent event – and give up its regular behavior.

In many RTOS implementations, the primary characteristic of a task is determined at compile time. In RTuinOS, however, this is done partly at compile time and partly at runtime. RTuinOS is provided as a single source code file which should be compiled together with all the other code so that it becomes an RTuinOS application. In the most simple case, if we do not define any task, the application will strongly resemble a traditional sketch, whit a setup and a loop function. The former will run only once, in the beginning, and the latter will run repeatedly.

G. FreeRTOS port for Arduino

FreeRTOS is a Real-Time Operating System (RTOS) designed to be small and simple. Its kernel consists of only three C files and provides few routines in Assembly which needs to be rewritten to any new ported architecture. FreeRTOS provides methods for multiple threads or tasks, mutexes, semaphores, and software timers. Thread priorities are also supported. It also provides a tick-less mode for low power applications. Tick-less is an approach in which timer interrupts do not occur at regular intervals, but are only delivered as required. FreeRTOS applications can be entirely statically allocated.

There are some FreeRTOS ports for Arduino such as the ones created by Stevens [21] and Greiman [22]. While both projects provide a similar approach, Steven's project [21] is more recent and came to activity in late 2018. It can be used on many Arduino models such as Arduino UNO, Mega, MCU based, and others. In this approach, tasks are created on the `setup` function and the main `loop` function is free to run any other specific routine.

Although this approach looks interesting, it increases storage usage, and the programmer needs to pay attention to the required space size. As an example, an empty sketch that with FreeRTOS packages included compiled with Arduino IDE v1.6.9 on Windows 10, makes use of 21% more memory space on an Arduino Uno and 9% more on a Mega when compared to a genuine empty sketch. While this approach is very powerful, it also requires minimal knowledge of operating systems, threads and synchronization.

H. Qduino

Qduino [23] is an operating system and programming environment developed to run on multicore x86 platforms and Arduino-compatible devices such as Intel Galileo. It provides support for real-time multithreading extensions to the Arduino API, which promises to be easy to use and allows the creation of multithreaded sketches, as well as synchronization and communication between threads.

Furthermore, Qduino intends to provide real-time features that provide temporal isolation, between different threads and asynchronous system events such as device interrupts, an event handling framework that offers predictable event delivery for I/O handling in an Arduino sketch. One of its main offers is being a platform with smaller memory footprint and improved performance for Arduino sketches and backward compatibility that allows the execution of legacy Arduino sketches.

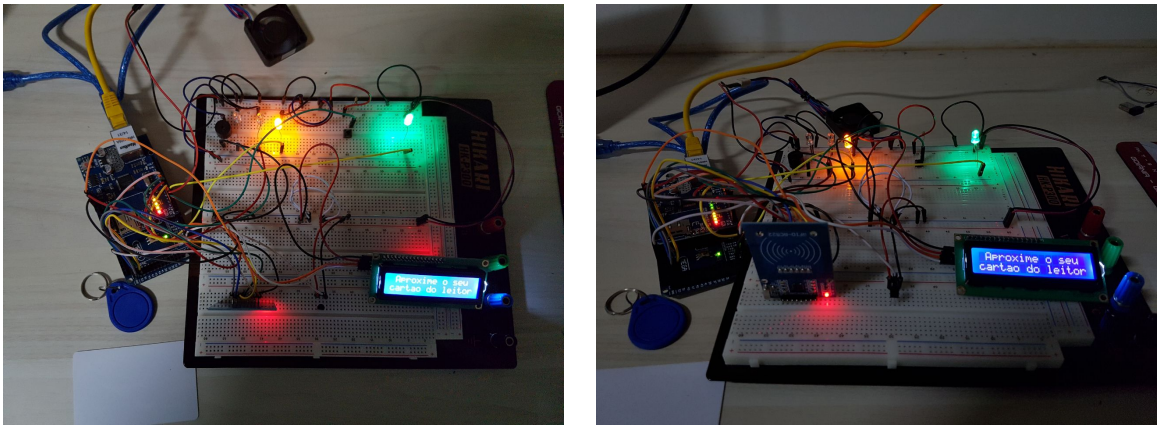
Although Qduino seems to be a robust approach for dealing with the multithreading issue, it is, unfortunately, not available for regular Arduino boards such as Arduino Uno or Arduino Mega.

III. Case Study based on Timed Interrupts

Due to the standard API using, low memory cost and simple implementation, we opted for analyzing the use of timed interrupts as an approach to allow multithreading on Arduino boards. We created a scenario for this study that requires dealing with external actions and input/output, reading a sensor and acting differently according to the read value, blinking of LEDs for a defined amount of time, showing information on a display, and continuously updating a web server based on information collected from the entire system. The system proposed and how its prototype was implemented is described next.

Our prototype provides an access control system based on RFID (radio frequency identification) [24], which simulates access granted or denied and gives feedback by turning on a green or red led for a defined amount of time while information about the access is being shown on an LCD display. To combine it with another external action to the system, we added a fan, which is turned on by pressing a push button, and whose speed is determined by the room temperature. The system also includes a web server, which can exhibit information about the place being secured and allows the system to be remotely monitored.

To build the prototype, we have used an Arduino Mega 2560 – a microcontroller board based on ATMega2560. It has 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button [25]. Besides powering up all components used, the Arduino Mega is used to receive and process all data and make the required decisions. For the access control system, we used an MFRC522 RFID reader with an operating frequency of 13.56MHz and maximum data transfer rate of 10Mbit/s [26]. A set of green, red and yellow LEDs were used for simulating access granted, access denied and awaiting card to be read, respectively. A buzzer was also used to emit a different frequency sound according to the access that was granted or not. This information is also shown on an I2C 16x2 LCD [27]. Figure 1 presents a view of the actual prototype developed.



(a) Top View (b) Perspective view.

Figure 1. System Prototype.

A. *Proposed Scenario*

In the system proposed, the fan was simulated with a 5V cooler, which was turned on/off with the press of a push button, and its speed was determined based on the temperature read by an LM35 sensor. Finally, the web server was possible thanks to an Ethernet shield, allowing the Arduino board to connect to the internet. The schematic of the prototype is shown in Figure 2.

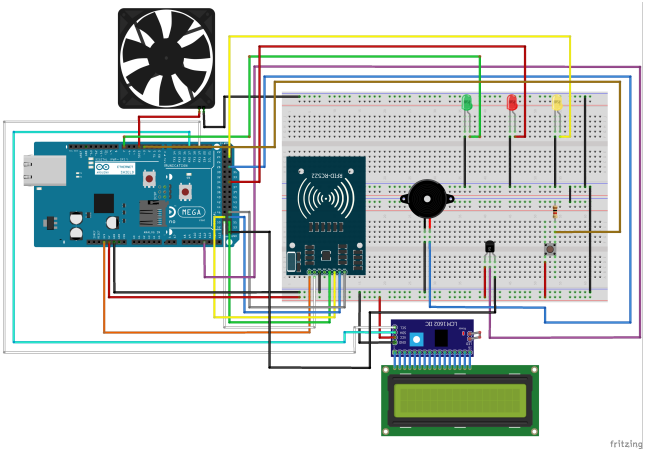


Figure 2. Circuit Diagram.

When the system starts, a yellow LED turns on, and the LCD display shows the message "Approximate your card". If an RFID tag is read, meaning a user has presented a card and wants to access the room being secured, the Arduino checks if the tag presented is assigned to a person who can access the room where the system is installed. If so, then the door of the room is open, letting the person walk in, and then closed shortly after, automatically. In our prototype, this is represented by a green led which is turned on for 10s, while the yellow LED is kept off. On the other hand, if a card is presented and the person assigned to that card is not in the list of authorized users or is not listed as a recognized user of the system, then the yellow LED turns off while a red LED is turned on for 5s.

When the time is up, the green LED turns off, and the yellow LED turns back on, indicating that the door has been closed and that a card needs to presented again in order to allow access to the room once again. Whenever a user presents a card to the RFID reader, the LCD display indicates whether the user whose card was read has been granted or denied access, and the buzzer emits a beep on different frequencies when the access has been granted or denied. At any moment, if a user accesses the web server, they will see the status of the door stated, i.e., "open" and "closed", the temperature of the room, and if the fan is on or not.

The fan installed at the room can be turned on or off whenever a button is pressed. When turned on, its speed is determined by the temperature of the room, collected periodically by the LM35 sensor. The value collected by the sensor is then passed to a map function that receives the minimum and maximum pre-set range of temperature carefully chosen to represent the minimum and maximum power, respectively. The temperature is continuously checked, powering the fan accordingly and updating the temperature value on the web server whenever a client is accessing the web server.

B. *Scenario Evaluation*

The scenario proposed combines many tasks which are user or state dependent. To deal with all required tasks, we provide a solution using interrupts and timed interrupts. In this section, we provide details on the solution provided and comment on its efficiency.

The solution proposed contains the `setup` function, the `loop` function, and a few other functions implemented to deal with each of the tasks required by the system. The `setup` starts the MFRC522, display, web server, interrupts, and defines the pins used as input or output.

The `loop` function is also straightforward. It is responsible for exhibiting information of "Approximate your card",

"Access granted" or "Access denied" on the LCD display and managing the changing of the status of the LEDs whenever a card is presented to the system, as well as performing the beep sound. All actions executed by `loop` are done based on the state changes of variables associated with the access control system.

To guarantee that a button press, meaning the user wants to turn on or off the fan, is never missed, we used an interrupt attached to the Arduino pin the button is connected. In this case, whenever a button press occurs, the system can detect it, and the state associated with the fan is changed to on/off. The task of effectively controlling the fan and its speed is handled by a timed function, which is executed by the system every 0.5s, no matter what other task is being handled in the system, and it acts based on the current state of the fan. The flow diagram of the function responsible for controlling the fan is illustrated in Figure 3.

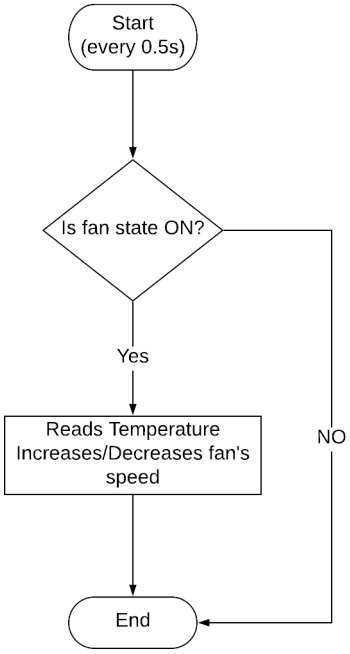


Figure 3. Fan flow diagram.

The access control is also implemented based on a timed interrupt. Every 0.5s, the system checks if a card is being presented and, in the affirmative case, the system verifies if its ID is associated with any card stored at the list of allowed users. The state variables associated with the RFID are then changed, and the loop function handles the process of opening or not a door (lightning a green or a red LED), based on the current state of those variables. Figure 4 shows the flow diagram for the critic part of the access control system.

Finally, the web server control is also controlled by a timed interrupt, triggered every 0.5s. Whenever the function associated with the web server control is called, it verifies if the server has an active client. If so, it exhibits all current

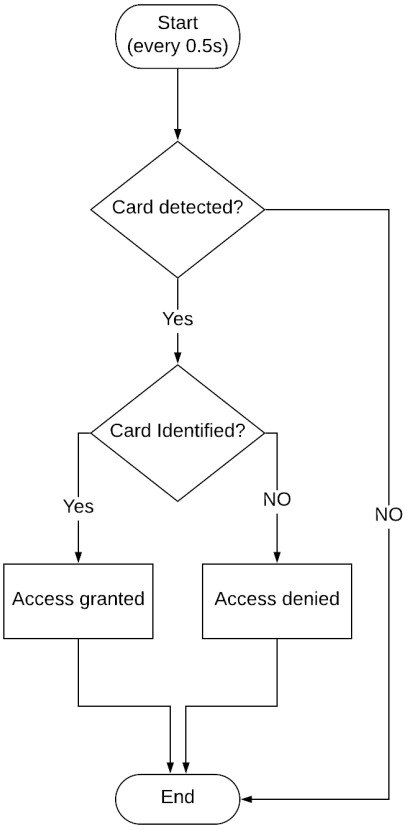


Figure 4. Access control flow diagram.

information about the system, including the status of the door (open/close/waiting for a card), the status of the fan (on/off), and the room's temperature. Since this function runs every 0.5s, the status presented on the web server is always up to date, with very little delay between the time a status was changed and the time it was first shown on the web page. Hence, whenever a card is presented to the system or the fan is turned on/off, the web server shows the card ID number that has been presented, shows if the door opened or remained closed, and shows the current state of the fan.

We empirically defined the time interval to 0.5s so that it does not affect the efficiency of the program nor does it waste processing time. The fan which has its speed defined according to the current temperature, for example, should not need to change its speed on a lower time basis, nor is the user interested in knowing the temperature of the room every millisecond.

IV. Discussion

In Section II, we described a variety of the available solutions for running or emulating multithreading on Arduino boards. We have seen that, while interrupts and timed interrupts are available by default on any Arduino, varying only the number of pins that can be used for such thing, all other

solutions described rely on external libraries. Furthermore, some of the solutions described are operating systems, such as Qduino, which is not available for the most common Arduino Boards, since it was planned to Intel Galileo Platform, or FreeRTOS, which although simple, consumes an enormous amount of memory. Moreover, no matter how simple a library can be, it requires an effort of reading and understanding how it works and how it can be useful for the desired application. We highlight that most of the solutions previously described, in essence, provide an abstraction of AVR timers to emulate threads.

We proposed a scenario in Section III that required dealing with many different external and user-dependent actions, and we evaluated the use of timed interrupts to deal with the proposed scenario. The solution proposed proved to be very simple to be implemented, it does not require any additional library as some of the approaches described in Section II, and it provided high efficiency in dealing with all the system's requirements, even though concurrent tasks have to be performed. We observed that the system could accurately identify when an authorized card was presented to the RFID reader, not missing a single card read trial, and thereafter allowing access for authorized users or denying it when an unauthorized card is used, without interrupting any ongoing task.

Moreover, the implemented web interface deserves highlighting, since it allows real-time verification of the status of the restricted environment where the system is installed and, as we show through the case presented for fan control, the system can be extended to display other information in the web system, such as the current ambient temperature and fan status, as described previously. An access history listing all individuals who had the entrance authorized, and even those who tried to enter the room and had the access denied, is another information that could be easily added to the web system with few modifications.

Since we used an Arduino Mega and it has 6 Timers, the proposed scenario fits adequately to deal with the system and user requirements. We note, however, that timed interrupts scheduled for the same interval can easily share routines, unless they perform blocking operations. Consequently, although only 6 Timers are available on Arduino Mega, much more than six time-dependent activities could be performed, as long as their routines are implemented together in the same timed interrupt.

V. Concluding Remarks

In this paper, we have gathered and described some software solutions for emulating threads on a single Arduino. From all software approaches presented, we chose timed interrupts, which can be implemented without the need of any additional library, and we analyzed its efficiency in dealing with many tasks concurrently. In order to evaluate how well timed interrupts could handle different actions, we proposed a case study of a real scenario that can be easily implemented and reproduced.

The case study proposed and evaluated required dealing with many different actions, including some user-dependent actions, such as input/output which could happen at unspecified time intervals. We proposed an access control system and implemented a prototype that required handling an RFID card, buttons, sensors, and synchronous blinking LEDs, besides continually updating the status exhibited on both an LCD and a web server.

Our solution based on timed interrupts requires no additional effort of adding a library since timed interruptions are part of the standard Arduino API. The solution presented to the case study proposed was able to handle all different tasks efficiently, even though blocking tasks and loop routines were being executed concurrently, while still maintaining the security and efficiency of the access control system. The software solution proposed shows that timed interrupts have the potential to be used in a wide range of applications, as well as could be commercially adopted.

Besides the simplicity of implementation, timed interrupts can help not only reduce the amount of Arduino boards in a project, consequently reducing the cost of a project, but it can also help to reduce the complexity of synchronization, connections among them, and so forth. We conclude that many areas that require the application of microcontrollers such as Arduino could benefit from this approach, since one could efficiently execute more functions with the inclusion of more components, such as GPS, sonar, gyroscope, and so on, all connected to a single Arduino. As a future work, we intend to compare the timed solution with some other available solutions, such as the ones described earlier in this paper, in order to adequately evaluate and compare the advantages and disadvantages of each solution.

Acknowledgments

The authors would like to thank CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil), CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brazil), IFG (Instituto Federal de Goias – Brazil) and CISTER Research Centre in Real-Time & Embedded Computing Systems – Portugal for their support. We would also like to express gratitude towards the anonymous reviewers whose valuable comments and suggestions greatly improved the quality of the paper.

References

[1] "Arduino homepage," https://www.arduino.cc/, 2019, accessed: 2019-01-20.

[2] Y. Pititeeraphab and M. Sangworasil, "Design and construction of system to control the movement of the robot arm," in *Biomedical Engineering International Conference (BMEiCON), 2015 8th*. IEEE, 2015, pp. 1–4.

[3] S. P. Deshmukh, M. S. Shewale, V. Suryawanshi, A. Manwani, V. K. Singh, R. Vhora, and M. Velapure, "Design and development of xyz scanner for 3d printing," in *Nascent Technologies in Engineering (ICNTE), 2017 International Conference on*. IEEE, 2017, pp. 1–5.

[4] Z. H. Soh, M. H. Ismail, F. H. Otthaman, M. K. Safie, M. A. Zukri, and S. A. Abdullah, "Development of automatic chicken feeder using arduino uno," in *Electrical, Electronics and System Engineering (ICEESE), 2017 International Conference on*. IEEE, 2017, pp. 120–124.

[5] M. Kamisan, A. Aziz, W. Ahmad, and N. Khairudin, "Uitm campus bus tracking system using arduino based and smartphone application," in *Research and Development (SCOReD), 2017 IEEE 15th Student Conference on*. IEEE, 2017, pp. 137–141.
[6] J. Toji, Y. Iwata, and H. Ichihara, "Building quadrotors with arduino for indoor environments," in *Control Conference (ASCC), 2015 10th Asian*. IEEE, 2015, pp. 1–6.
[7] J. Sarik and I. Kimiss, "Qduino: A multithreaded arduino system for embedded computing," in *Frontiers in Education Conference (FIE), 2010, IEEE*. IEEE, 2010, pp. T3C–1–T3C–5.
[8] S. Jindarat and P. Wuttidittachotti, "Smart farm monitoring using raspberry pi and arduino," in *International Conference on Computer, Communications, and Control Technology (I4CT), 2015*. I4CT, 2015, pp. 284–288.
[9] A. Garrigos, D. Marroqui, J. Blanes, R. Gutierrez, I. Blanquer, and M. Canto, "Designing arduino electronic shields: Experiences from secondary and university courses," in *Global Engineering Education Conference (EDUCON), 2017, IEEE*. IEEE, 2017, pp. 934–937.
[10] A. S. Tanenbaum, *Modern operating system*. Pearson Education, Inc, 2009.
[11] "Arduino reference," https://www.arduino.cc/reference/en/, 2019, accessed: 2019-01-20.
[12] "Atmega328/p datasheet," http://ww1.microchip.com/downloads/en\/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf, 2016, accessed: 2019-01-20.
[13] "Atmel atmega 640 / v-1280 / v-1281 / v-2560 / v-2561 / v datasheet," http://ww1.microchip.com/downloads/en/\DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf, 2014, accessed: 2019-01-20.
[14] I. Seidel, "Arduino thread," https://github.com/ivanseidel/ArduinoThread, 2013.
[15] C. Moos, "avr-os," https://github.com/chrismoos/avr-os, 2013.
[16] A. Dunkels, O. Schmidt, T. Voigt, and M. Ali, "Protothreads: Simplifying event-driven programming of memory-constrained embedded systems," in *Proceedings of the 4th international conference on Embedded networked sensor systems*. Acm, 2006, pp. 29–42.
[17] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*. IEEE, 2004, pp. 455–462.
[18] "Protothreads," http://dunkels.com/adam/pt/index.html, 2019, accessed: 2019-01-20.
[19] P. Vranken, "Rtuinos," https://svn.code.sf.net/p/rtuinos/code/trunk/doc/\doxygen/html/index.html, 2017.
[20] ——, "Rtuinos," https://github.com/PeterVranken/RTuinOS, 2013.
[21] P. Stevens, "Arduino freertos library," https://github.com/feilipu/Arduino_FreeRTOS_Library, 2017.
[22] B. Greiman, "Freertos arduino library," https://github.com/greiman/FreeRTOS-Arduino, 2013.
[23] Z. Cheng, Y. Li, and R. West, "Qduino: A multithreaded arduino system for embedded computing," in *Real-Time Systems Symposium, 2015 IEEE*. IEEE, 2015, pp. 261–272.
[24] S. Ahuja and P. Potti, "An introduction to rfid technology." *Communications and Network*, vol. 2, no. 3, pp. 183–186, 2010.
[25] "Arduino store," https://store.arduino.cc/usa/arduino-mega-2560-rev3, 2019, accessed: 2019-01-20.
[26] "Mfrc522 datasheet," https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf, 2016, accessed: 2019-01-20.
[27] "Datasheet i2c 1602 serial lcd module," https://opencircuit.nl/ProductInfo/1000061/I2C-LCD-interface.pdf, 2019, accessed: 2019-02-23.

SESSION 2

AI and Machine Learning

Survey on Explainable Artificial Intelligence (XAI)

Leonardo Ferreira
Faculty of Engineering, University of Porto,
Portugal
up201305980@fe.up.pt

Abstract— **In the last decade, with the increasing computation power and data available, Artificial Intelligence and Machine Learning have significantly improved several systems such as face and speech analysis, decision making, and others. Unfortunately, most of the AI approaches available use models that are complex and difficult to explain such as black boxes. This lack of transparency and interpretability is a big drawback for society. Medicine, finance and military are some examples of fields where transparency is a critical and vital requirement to build trustworthy systems. Thus, there is a growing interest on knowing how to understand and explain these intelligent procedures in both academia and in the industry.**

This survey contextualizes the explainability problem by addressing some of the most important key aspects involving this concept and summarizes a few works that we have explored in this area. First, we provide some notions and the relevance that this technology might have for the AI future. Then we pinpoint some XAI (Explainable AI) issues such as ethical and lack of transparency problems, and taxonomies available. Through related works, we review existing interpretation methods that can extract or improve the explanation's level. Existing XAI systems are also explored.

Keywords: Explainable AI, Machine Learning, Transparency, Interpretability, Black Box Systems.

I. INTRODUCTION

Over the last decade, Artificial Intelligence (AI) powered by Machine Learning (ML) and Deep Learning has suffered several changes in order to perform various tasks such as face recognition and behavior prediction. Currently, we can say that AI handles a lot of decisions in our daily lives. However, researchers and companies have begun to push machine learning to other "higher" fields that handle more critical and sensitive decisions like healthcare, banking, military an security [1]. The International Data Corporation (IDC) foresees that global investment on AI will grow from 24 billion U.S. dollars in 2018 to 77.6 billion U.S. dollars by 2022 [2]. Regardless of these facts, there is also a growing concern around the use of these AI systems. As the importance of the decisions aided using machine learning increases, it becomes more important for users to be able to suitably weight the assistance provided by such structures. However, some trust issues and problems regarding giving explanations arise from these frameworks. The way by which these systems achieve their choices isn't in every case clear, particularly when they use machine learning strategies to predict certain behaviors. For example, in life-changing decisions such as cancer diagnosis, it is important to understand the reasons behind such a critical decision. Machine learning algorithms, more specifically black boxes, present excellent results and predictions but it is hard to get insight about how they reach certain decisions. This is where Explainable, Interpretable or Transparent AI plays an important role. Explainable AI aims to build/improve techniques that create more "transparent" models without sacrificing performance.

With these concerns in mind, the present article shows a survey on Explainable AI (XAI). There are other recent surveys that talk about some characteristics presented by this topic and do a state of the art in depth such as in the surveys [3] and [4]. This survey also provides knowledge about some of the most used explainability methods but considers another important aspect: systems/products that use this concept on their policies to analyze real world data.

In this sense, we believe this survey can lead researchers and companies to a path towards promising and suitable directions. Unlike the studies mentioned, we also advocate some XAI architectures that can help current and future companies optimizing their services through more transparent and authentic systems.

The remainder of this paper is organized as follows. In section II, we address some general concepts concerning Explainable AI. Section III presents related works in terms of algorithms and systems. Section IV discusses some of the problems collected in the literature review and then, Section V concludes the survey.

II. GENERAL CONCEPTS

XAI is a field of AI that seeks to turn AI systems more transparent and understandable to users.

Despite being a term that is rather new, the explainability problem emerged a long time ago (around the mid-80s) when trying to explain expert systems [5]. For many years, priority has been given to the performance over the interpretability leading to huge advancements in several fields including computer vision and natural language processing. However, since two years ago, AI research has shifted towards a whole new dynamic where explainability may be among the key measures for evaluating models. This change resulted from the influence that AI and ML had over the industry and also because there was a need for more detailed information when dealing with more critical decision making processes.

DARPA (Defense Advanced Research Projects Agency) [6], defined two missions for Explainable AI:

- Generate more explainable models without compromising that accuracy achieved, which in some cases may be hard to get since there is a trade-off between explainability and performance;

- Deliver logical reasons to users, so they can trust and manage these new and innovative methodologies. Questions such as "Why did you do that?", "Why not something else?", "When do you succeed?", "When do you fail?", "When can I trust you?" and "How do I correct an error?" are some examples that XAI systems must be able to respond.

If society is able to trust in these new and complex artificial intelligence systems many areas can benefit from them. Nonetheless, as shown by Goodman and Flaxman [7] this kind of operations brings a lot of legal and ethical questions to discussion. These may restrict explainable AI systems in several industries but they can also conduct to new frameworks and opportunities capable of fighting discrimination.

A. XAI Importance

Explaining a certain action or decision is very important not only when we want to establish a relationship of trust but also to comprehend the reason behind it. Having an accurate model might be good but having explanations is better. XAI carries out various challenges.

1) Improve the system's performance [8]: the analysis of the performance and behavior of black boxes models can improve them. The introduction of XAI may also help on explaining why a model can be more efficient than other. The better we understand the problem or what a model is doing the easier it gets to improve it;

2) Learning new evidences and patterns [8]: AI systems analyze a great amount of data, being able to identify patterns which cannot be observed by humans (we can only process small amounts of data). These patterns can show us new behaviors that may explain why a certain course of action was taken;

3) Cover more areas such as legislation [8]: with the addition of AI structures in our lives it is important they follow the laws that guide society. Since it is not possible to rely on black boxes when dealing with legal aspects, it is imperative that the AI becomes more explainable. For example, imagine someone would like to ask for a bank loan and the bank, equipped with specialized Machine Learning algorithms, denies the request. Certainly the person would like to know the reason(s) for the denied appeal. Only explainable AI can guarantee information about the decision made.

Fig. 1 demonstrates how Explainable AI can be a very important tool, not only as a provider to a more trustworthy AI but also on helping to solve problems in several areas (explored later).

B. XAI challenges

Despite these advantages, Explainable AI will always face ethical, regulatory and business-critical issues around how we use the output given by the machine learning model.

There are some cases where it is very hard to get and understand the reasons that lead to certain outcomes. For example, when someone plays Go (board game), they can explain their decisions but DeepMind's AlphaGo[1] couldn't, at least in a way that is fully understandable to humans. In this case the fact that decisions can't be explained is not critical because it does not represent any kind of danger. However, in other fields developing and deploying AI systems that we cannot understand could lead to potential harmful situations, specially the ones that involve sensitive and critical data.

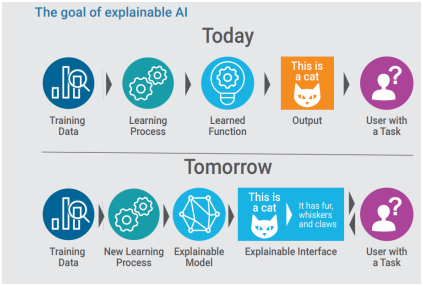


Fig. 1: Present AI vs Future AI [9]

There is another challenge to consider that can slow down explainability. Not all AI can be made explainable, and when it can, it can result in inefficient systems and hard choices. Businesses that already have AI models in use, will have a hard time to turn them explainable (time and money spent will be big). And, as mentioned before, some companies are still unwilling to change since many questions are still unanswered (legal and ethical are the most difficult ones).

C. XAI Objectives

According to Lipton [10], explainability uses the concepts:

1) Trust: one of the major concerns of using explainability in AI is how much we can rely on these kind of systems;

2) Causality: researchers study the possibility of inferring properties or generating hypotheses about the real world with a learnt model (trained before);

3) Transferability: related to the models capacity to extend and generalize its properties to other different environments from the ones it was trained on;

4) Informativeness: instead of using decision theory only on the supervised models outputs to act without the human intervention, it can be used to assist in the decision making;

5) Fairness: how can we verify that the decisions made by the AI are fair? To answer this question, the fair concept must be well defined. The rules followed by the model must comply with the ethical principles [7].

There are other authors like Doshi-Velez and Been Kim that defend more goals [11] that despite not being in the same level as the ones presented previously they are also pertinent. These are privacy (data is protected) and robustness (small changes in the inputs don't change dramatically the output).

D. XAI Taxonomy

Since this is an area that is constantly changing, multiple XAI taxonomies have been exposed by the research community.

[1] https://deepmind.com/research/alphago/

Lipton [10] studied the concept of how to achieve interpretability and managed to come up with an idea of how to represent it. Christoph Molnar [12] made a similar classification of interpretable/explainable methods. Yao Ming [13] also tried to represent explainability based on model-unaware and model-aware explanations. All these studies are almost identical (the difference resides on the techniques allocation and some other properties). Table I shows two types of interpretability/explainability.

TABLE I: XAI Taxonomy

Category		**Model Techniques**
Intrinsic		Linear Models, Logistic Regression, Decision Trees, KNN, among others.
Post-hoc [1]	**Local**	Approximation
		Feature Importance
		Visualization
		Text Explanations
		Explanation by Example
	Global	Approximation
		Feature Importance
		Visualization
		Neuron Inspection

[1] **Can be model-specific or model-agnostic. (Further detailed in Section III).**

With Intrinsic mechanisms it is possible to explain the predictions through the model itself (models like Linear models, Naive Bayes, Logistic Regression, decision trees and others). It is composed by three transparency properties:

- Simulatability - a human is capable of gathering the training data together with the model parameters and process in a reasonable amount of time all the steps required until the model produces a prediction;
- Decomposability - every part of the model such as each input, parameter and calculation is understandable to the user;
- Algorithm transparency - the learning algorithm is understood and assured to work on a future new problem.

On the other hand, there are Post-Hoc approaches where explanations are given after applying interpretation methods on black box models. They can be classified by scope or by model. By scope, it is evaluated as being global or local whereas by model, it can be model-specific or model-agnostic. On local a single prediction is explained while in global an entire model or set of predictions are interpreted. Depending on the nature of the problem and the level of explanation required (local or global), explanations can be generated by approximation, feature importance, visualization, text explanation feature importance, text explanations, explanation by example and neuron inspection techniques. Model-specific techniques are specific to models (all intrinsic methods are model-specific) while model-agnostic tools can be used on any machine learning model. Some of these techniques will be discussed in further detail in section III.

E. Main Applications areas

Knowing how much we can benefit from explainable AI is difficult to predict. Nowadays, the market is trying to become more intelligent and sophisticated through the ability of providing reasonable and accurate explanations.

Explainability has been gradually introduced in several markets such as Finance and Banking (learning finance and banking behaviors could lead to an improvement in resources management), health care (medically-validated tools for predicting surgical outcomes and cancer mortality can have a tremendous impact), insurance (automated claims processing and insurance claimant segmentation), other areas such as cybersecurity, military, among others.

III. STATE OF THE ART

First some existing surveys are presented and then Intrinsic and Post-Hoc explainability methods are explored. Post-Hoc methods will be more emphasized since they are the ones that best handle black boxes models. Examples of companies/products that use this concept in their architectures will also be analyzed.

A. EXISTING SURVEYS

Explainable AI had a huge growth in the past years and as a result, some research works have been published.

Despite Lipton's work [10] not being a survey, it provides a robust breakdown about what might constitute interpretability through some available techniques.

Guidotti et al. [14] presented a survey where they review several methods for explaining black boxes. They showed a well structured and detailed taxonomy of explainability methods that depend on the type of problem and data confronted. However, they only focus on interpretability mechanisms, discarding other important metrics such as evaluation.

Doshi-Velez and Been Kim published a work [11] where they state a possible taxonomy for explainability and the best practises to take the most advantage of this topic. They don't get into much detail about existing methodologies but they do a deep reflection regarding explainability measurement.

As the last study, Dosilovic et al. [15] suggested an overview about explainability and presented some improvements over machine learning models under the supervised learning paradigm, with Deep Neural Networks (DNN) as the main target.

The literature review that is going to be presented next by this article was achieved by looking into papers from four databases: IEEExplore, SCOPUS, Google Scholar and ACM Digital Library. The ArXiv repository was also needed. Keywords such as "explainable AI", "interpretability", "black boxes", "transparency" were used to facilitate the search for papers. This research was restricted to publications between 2004 and 2018. The selected papers were then evaluated based on the titles, abstracts and keywords to determine which were relevant articles for further analysis.

We will look into a few methods that can increase the level of explanations given and then some existing XAI systems.

Since it is almost impossible to study the 381 existing papers, only a subset of this works that we believe as being relevant, are going to be detailed.

B. INTRINSIC EXPLAINABILITY METHODS

Intrinsic models can be interpreted using the components model (simulatability), algorithm (algorithm transparency) and data (decomposability).

Simulatability states that a model is transparent or a "glass" model if it is understandable to humans. It is only achievable if the trained model is small and simple to comprehend. Usually, complex models have low simulatability because they include a great amount of data and processes that are impossible for humans to discern.

V. Krakovna and F. Doshi-Velez [16] suggested a method to increase the model interpretability through the combination of recurrent neural network (RNN) and hidden markov models (HMM). This requires the training of a HMM on LSTM (long short-term memory) states; a hybrid model where a LSTM is trained and augmented by HMM state distributions and a joint hybrid model where LSTM and HMM are trained simultaneously.

For Decomposability all the components of a model such as inputs/features, parameters and calculations must be transparent.

Concerning the input, Wu et al. [17] created a method named activation regularization, that encourages the activation function outputs to satisfy a target pattern which improves the network interpretability and regularization. This technique starts by rearranging the activation function outputs in a grid layout followed by normalization. Then the regularization function is applied to decrease the distance between the activation and the target pattern.

For parameters, Choi et al. developed a method named Reversed Time Attention Model or RETAIN [18]. It was developed to help doctors comprehend why a model was predicting patients to be at risk of heart failure. This method works with two neural networks which are analyzed with attention mechanism to have a better understanding of each neural network. After applying RETAIN, two new parameters are delivered: alpha and beta which turn the model more accurate and interpretable.

Regarding the calculations, Springenberg et al. established an approach [19] where all the layers of a CNN (convolutional neural network) are convolutional. This approach starts by substituting all the max-pooling layers (common in most CNN models) with convolutional layers through dimensionality reduction (via strided convolution) without losing accuracy on several image recognition benchmarks. Without max-pooling layers, the subsequent DeconvNet does not need switches, thus not affecting the input image.

Algorithm transparency not only gives us a better understanding of how the algorithm works, but also guides us to build and improve the models used.

Ross et at. [20] suggested a method to counter the problems presented by black box explanations tools (don't scale to explain entire datasets) that works with basis on right for the right reasons (RRR) rather than learning correlations between the data. This method verifies if an input feature is relevant or not to the classification of an example through binary masks, as assessed by a human expert. By using domain expertise in the models, the predictions are dependent on important features.

C. POST-HOC EXPLAINABILITY METHODS

A learned model can provide local or global explanations. Inside these are several interpretation techniques which can be model-specific or model-agnostic. It is important to note that, even though there are approaches developed to explain specific models (for example, neural networks), they can be adapted to explain any kind of model through the introduction of variations and improvements.

Model-agnostic tools are more flexible than model-specific approaches because the user is free to use any kind of machine learning methods. Next, some model-agnostic approaches are addressed.

Ribeiro et al. describe LIME [21], a local interpretable model-agnostic capable of providing explanations. It describes the predictions of any classifier or regressor by learning an interpretable model locally around the prediction. To understand why an output was produced by a machine learning model, the input given to the model is changed so we can observe how the prediction changes. Then, LIME produces a new dataset consisting of changed samples and the associated black box models predictions so he can train an interpretable model weighted by the proximity of the sampled instances to the instance of interest.

Fig. 2 is an example of how LIME can provide explanations. It shows the area which is most likely for an image to contain an electric guitar, acoustic guitar or the dog Labrador.

Fig. 2: LIME explanations [21]

Friedman proposed the Partial Dependence Plot (PDP) [22], a method that provides the visualization of the impact that each independent feature on the model predictions values has after averaging all other features used by the model. It is considered a global method since it takes into account all the available instances and reports about the global relationship of a single feature with the predicted result.

Another PDP variation was shown by Goldstein et al., the Individual Conditional Expectation (ICE) [23]. Instead of having one single line for the predicted response on a feature, ICE analyzes each instance separately, resulting in multiple lines, one for each instance. The average of ICE plots corresponds to PDP. Despite only displaying one feature at a time they are more intuitive to understand, in

most cases it is recommended to use ICE plot with PDP to gain more insight over the data.

Apley showed an optimized version of the PDP called Accumulated Local Effects (ALE) Plot [24]. ALE and PDP reduce the function by averaging the effects of the other features, however they diverge in whether averages of predictions or of differences in predictions are calculated. They are also different on whether averaging is performed over the marginal or conditional distribution. While PDPs average the predictions over the marginal distribution, ALE averages the changes in the predictions and accumulates them over the grid. ALE plots are unbiased, which means they still work when features are correlated (PDP fails in this scenario), faster to compute and their interpretation is more clear.

Altmann et al. described the Permutation Variable or Feature Importance [25]. This strategy tries to measure the model's prediction error by changing one feature's values in order to see if a feature is important or not to the model. A feature is relevant if the model relies on the feature for the prediction and if changing its values increases the model error. A feature is irrelevant if the model ignores the feature for the prediction and if changing its values keeps the model error unchanged. ALE plots provide global and local explanations depending on the problem.

Montavon et al. reported a similar technique that quantifies the importance of each feature called Sensitivity Analysis [26]. It justifies the model's prediction based on the model's locally evaluated gradient which is partial derivative or other local measure of variation. In other words, it checks what is/are the features that are more susceptible to the end result. This technique also gives global and local explanations depending on the problem at stake.

Lundberg and Lee designed another methodology called Shapely Explanations [27]. This approach computes feature contributions for single predictions with the Shapley values which is a concept in cooperative game theory. The features values of an instance cooperate to achieve the prediction. The Shapley value fairly distributes the difference of the instance's prediction and the datasets average prediction among the features.

Kim et al. proposed a method named Prototypes and Criticisms [28]. This example-based technique depends on prototypes (most important instances in the data) and criticisms (catches the instances missed by the prototypes). In order to use them, it is imperative to find them in the data. For that the MMD (Maximum Mean Discrepancy) critic framework is required. This tool needs the number of prototypes and criticisms as input, a kernel function to estimate data densities, a MMD to tell us how different two distributions are and a witness function to show us how different two distributions are at a particular instance. After that, it applies greedy search to locate the prototypes and the criticisms. Supplies the user with local or global explanations depending on the situation.

Wachter et al. presented Counterfactual explanations [29]. This method describes as minimally as possible the conditions that produced a certain outcome without entering too much on the algorithm decision itself. They are used to explain predictions of predicted instances. Unlike prototypes, counterfactuals don't have to be actual instances from the dataset, but can be a combination of feature values.

On the contrary, model-specific methods are dependent on the nature of the machine learning model used. Most of the techniques present in this area are used to interpretate predictions of neural networks and can give local/global explanations depending on the case at hand.

Bach et al. presented Layer-Wise Relevance Propagation (LRP) [30], one of the main model-specific algorithms that explains decisions/predictions by decomposition. The prediction achieved is redistributed backwards through the neural network from the observation until it gets a relevance score from the input variable. The neurons that contribute the most to the higher-layer receive most relevance from it. It has the goal of finding the best relevance score R over the input features such that the sum of the value of R represents the network output.

Fig. 3 shows the output after applying LRP. The highlighted pixels (the white ones in the right image) represent the most relevant pixels that indicate the presence of a dog.

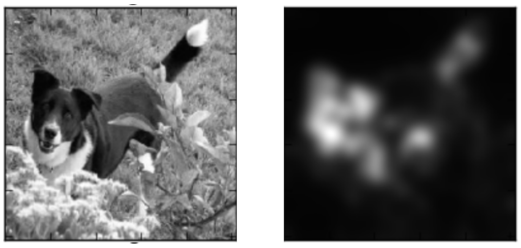


Fig. 3: LRP Explanations [31]

Zeiler et al. demonstrated Deconvolution [32], also an optimized version of backpropagation but different from the guided variation. Instead of removing the negative gradients, deconvolution uses max-pooling layers (down-sample an input representation that can be image, hidden-layer output matrix and others), reducing its dimensionality and allowing assumptions to be made about features contained in the other regions. With max-pooling layers, it is possible to find the best propagated signal location in the image.

Fig. 4 represents the outputs that deconvolution delivers after taking an image as input. With the application of max-pooling layers, it can assume several features, each one in each layer.

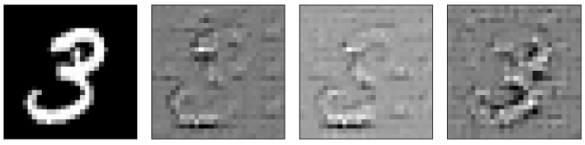
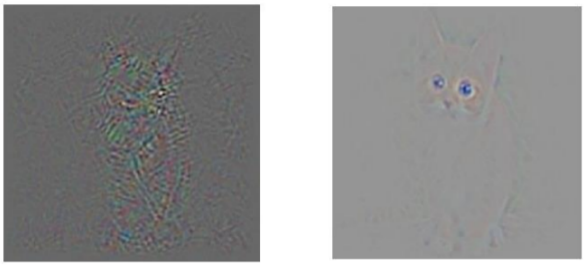


Fig. 4: Deconvolution [33]

Finally, Springenberg et al. suggested Guided Backpropagation [19], an optimized version of the regular backpropagation method. The classic backpropagation approach has the goal to analyze what happens to the cost function when there are changes in the weights and bias. In order to know the gradient of the cost function, the errors in the neural network are computed in a backwards manner (because cost is a function of outputs returned by neural network) from the final layer until it reaches the input layer. Guided propagation is better than the regular backpropagation because it ignores the negative gradients (negative influence to the cost function) by adding a guidance signal from the higher layers to the regular approach. Being guided, backpropagation only takes into account the observations that contribute positively to the class scoring.

As illustrated in Fig. 5 the results are cleaner with guided backpropagation when compared to the regular method due to the suppression of the negative gradients (noise).



Backpropagation Guided Backpropagation

Fig. 5: Backpropagation vs Guided Backpropagation [34]

All these approaches and algorithms represent ways to increase the level of knowledge extracted from data. There are other methods available that weren't included here (such as decision trees, rule based algorithms, decomposition, among others), since they are well detailed in the survey [3].

D. XAI Systems

XAI is still in their early stages and because of that there aren't many applications that use interpretability in their predictions. However, a few tools that try to incorporate this concept can be found in the market (some of them serve as a proof of concept and need more development).

Bertsimas et al. presented two frameworks that use explainability for the medical field [35] and [36].

The first one is called Potter (Fig. 6) [35] and it is an health care application that calculates the surgery risk. Potter uses an adaptive heuristic, the optimal classification trees (OCT) which allows more accuracy on the results obtained. After querying 382,960 patients, Potter presented a mortality c-statistic of 0.9162, a higher mortality c-statistic when compared to other associations like the American Society of Anesthesiologists (ASA) (0.8743), Emergency Surgery Score (ESS) (0.8910), and American College of Surgeons (ACS) (0.8975).

Bertsimas et al. created another framework called Oncomortality (Fig. 7) [36]. This tool is a web-based application capable of predicting the risk of mortality in cancer (all kinds) patients. Just like Potter, it also adopts classification trees. The feature importance method was also used so there was the possibility of knowing what were the features that matter the most for mortality prediction. This tool found out that 23,983 patients had a survival rate around 514 days. The prediction models used showed a better estimation quality on unseen data when compared to other models such as benchmarks (from 0.83 to 0.86) and identified the weight and albumin (protein in blood plasma) as being the most important features to the models.

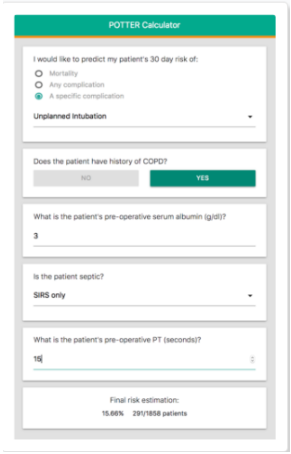


Fig. 6: Potter [37]

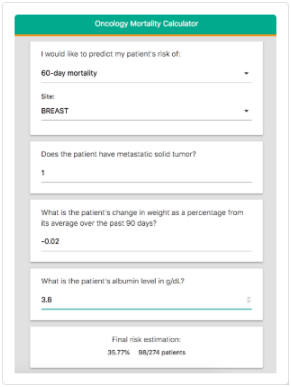


Fig. 7: Oncomortality [37]

DarkLight (Fig. 8) [38] is another XAI cybersecurity expert system that uses transparent and defendable logic to ensure better and safe experiences. DarkLight is understandable because it uses the same domain specific concepts that cybersecurity professionals use daily. It has several features such as being an AI expert system (ability to emulate decision making), uses ontologies instead of rules, supports a Scientific Foundation to cybersecurity among other features.

Fig. 8: DarkLight Process [38]

DarkLight uses sensing which maps instances from various sources and maps them to a model; Sense Making to distribute what is known and interpretate the data; Decision Making to emulate decision making and what is the best course of action; and acting to plan the response. This process flow allowed DarkLight to improve cybersecurity analysts by over one hundred times, reduce false-positives by up to 99% and boost return on existing cybersecurity investments.

SimMachines (Fig. 9) [39] is a machine learning and AI company also focused on the topic explainability. Their main areas of interest are finance, marketing, media and retail.

SimMachines uses a proprietary similarity-based machine learning to outperform other approaches in terms of speed and precision, while providing the justification behind each prediction. This allows them to handle data in its native form, significantly improving accuracy and giving them the ability to easily handle both structured or unstructured data. They further use a dynamic dimension reduction technique to identify the variables required to make an accurate prediction. As a result, they are able to provide the input variables that most strongly influenced the prediction, providing transparency to the process.

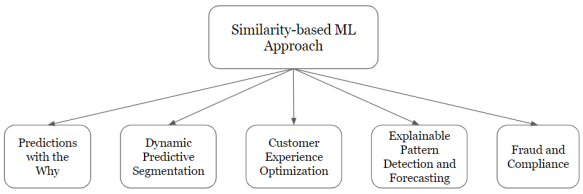


Fig. 9: SimMachine Stages [39]

In each stage, SimMachine:

- Provides insights into what factors drove the prediction;
- Generate dynamic predictive segments by grouping similar predictions together to enable the analysis of the machine-driven factors behind each segment ;
- Compare segments, trend segments over time, and forecast the future behavior of a segment;
- Improve the customer interaction with more precision and speed;
- Enable businesses to detect emerging patterns, forecast future demand or consumption, and view historical trends in granular depth;
- Detect fraud threats through fraud algorithms.

In the future, it is expected that there is an increased usage of explainability due to all the advantages it can offer.

IV. DISCUSSION

The search towards a more explainable AI has been gathering momentum for some time. Systems based on trust and fairness are significantly more powerful and efficient when compared to traditional ones. Nonetheless, and as demonstrated before, there are multiple questions that must be addressed.

First there is no consensus about explainability in the context of supervised learning. Even though the research community uses the terms "Interpretability" and "Explainability" synonymously it is important to establish a clear classification, specially when moving beyond systems that merely predict events to structures capable of supporting interventions. The work [4] suggests a distinction between these concepts. It differentiates them as the interpretation arrived at by a user agent when given an explanation from a model.

With the rapid growth of AI systems there have also been legitimate concerns about the intention (positive or negative) of this technology. Making XAI more accountable while facing the ethical, political, and legal challenges is very important, especially with GDPR implemented, where companies which cannot provide an explanation and record as to how a decision has been reached are heavily penalized (whether by a human or computer). Many industry sections (finance, medical and military) suspect that explainability may lead to discriminatory and dangerous decisions.

Another problem illustrated in the state of the art is the lack of a common explainable framework. Such a tool could help producing a variety of interpretablity methods with the capability of handling several types of data. With it, we could possibly combine existing interpretability methods to achieve a more powerful and complete explanation. Indeed, in the literature review we have seen some techniques that can complement each other (e.g. sensitive analysis and visualization).

The user and its understanding when towards an explanation also needs more attention. Providing simple and accurate explanations is not enough. The user, which could be a human or machine, must be able to understand them. As shown by Kahneman [40], humans behave, think and act differently so, clearly, different interpretations can arise after looking into the model's explanations. Anticipating interactive explanation systems that support many different reactions after presenting an explanation to the user, is a research path that needs more work. In fact there aren't many studies (almost none) that consider these two parameters (the explanation model and the user) and the works that exist are too isolated to even consider them as a valid option.

V. CONCLUSIONS

Explainable AI is an important but sometimes overlooked characteristic for a lot of systems. It complements and boosts decisions, yet brings to the surface many critical and problematic questions (ethic and quality of life implications). For this survey, we reviewed some of the most relevant interpretability techniques that provide explanations to users and some business or proof of concept systems that consider or are starting to incorporate these concepts on their architectures and products.

As shown by this paper, there is a considerable amount of work done over this concept; however, most of the related work focuses only on the explanation side, discarding completely the importance of the user understanding the same. Furthermore, the explainability concept is not studied by a standalone research community. Instead it is performed by different research groups, wasting the opportunity to find a common and integrated way to develop an ethical and transparent AI.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[2] International Data Corporation (IDC), "Worldwide Spending on Cognitive and Artificial Intelligence Systems Forecast," Available at https://www.datanami.com/2018/05/30/opening-up-black-boxes-with-explainable-ai/ (accessed at 27th October 2018), September 2018.

[3] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[4] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, M. Srivastava, A. Preece, S. Julier, R. M. Rao, T. D. Kelley, D. Braines, M. Sensoy, C. J. Willis, and P. Gurram, "Interpretability of deep learning models: A survey of results," in *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug 2017, pp. 1–6.

[5] W. R. Swartout and J. D. Moore, "Explanation in expert systems: A survey," Univ. Southern California, Los Angeles, CA, USA, Tech. Rep. ISI/RR-88-228, 12 1988, p. 58.

[6] Defense Advanced Research Projects Agency (DARPA), "XAI program," Available at https://www.darpa.mil/ (accessed at 27th October 2018), March 1996.

[7] B. Goodman and S. Flaxman, "European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"," *AI Magazine*, vol. 38, pp. 50–57, 2017.

[8] W. Samek, T. Wiegand, and K. Müller, "Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models," *CoRR*, vol. abs/1708.08296, 2017. [Online]. Available: http://arxiv.org/abs/1708.08296

[9] Datanami, "Opening Up Black Boxes with Explainable AI," Available at https://www.datanami.com/2018/05/30/opening-up-black-boxes-with-explainable-ai/ (accessed at 27th October 2018), May 2018.

[10] Z. C. Lipton, "The Mythos of Model Interpretability," *Queue*, vol. 16, no. 3, pp. 30:31–30:57, Jun. 2018. [Online]. Available: http://doi.acm.org/10.1145/3236386.3241340

[11] F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," *arXiv e-prints*, p. arXiv:1702.08608, Feb. 2017.

[12] C. Molnar, "Interpretable Machine Learning. A Guide for Making Black Box Models Explainable," Available online at https://christophm.github.io/interpretable-ml-book/ (consulted at 28/10/2018), 2018.

[13] Y. Ming, "A Survey on Visualization for Explainable Classifiers," 2017.

[14] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, "A Survey Of Methods For Explaining Black Box Models," *arXiv e-prints*, p. arXiv:1802.01933, Feb. 2018.

[15] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2018, pp. 0210–0215.

[16] V. Krakovna and F. Doshi-Velez, "Increasing the Interpretability of Recurrent Neural Networks Using Hidden Markov Models," *arXiv e-prints*, p. arXiv:1611.05934, Nov. 2016.

[17] C. Wu, M. J. F. Gales, A. Ragni, P. Karanasou, and K. C. Sim, "Improving Interpretability and Regularization in Deep Learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 256–265, 2018.

[18] E. Choi, M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun, "RETAIN: An Interpretable Predictive Model for Healthcare Using Reverse Time Attention Mechanism," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 3512–3520. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157382.3157490

[19] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," *arXiv e-prints*, p. arXiv:1412.6806, Dec. 2014.

[20] A. Slavin Ross, M. C. Hughes, and F. Doshi-Velez, "Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations," *arXiv e-prints*, p. arXiv:1703.03717, Mar. 2017.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 1135–1144. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939778

[22] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, vol. 29, pp. 1189–1232, 2001.

[23] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation," *Journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, 2015. [Online]. Available: https://doi.org/10.1080/10618600.2014.907095

[24] D. W. Apley, "Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models," *arXiv e-prints*, Dec. 2016.

[25] Altmann, Andr and Toloşi, Laura and Sander, Oliver and Lengauer, Thomas, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btq134

[26] G. Montavon, W. Samek, and K.-R. Müller, "Methods for Interpreting and Understanding Deep Neural Networks," *arXiv e-prints*, p. arXiv:1706.07979, Jun. 2017.

[27] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *arXiv e-prints*, p. arXiv:1705.07874, May 2017.

[28] B. Kim, R. Khanna, and O. Koyejo, "Examples Are Not Enough, Learn to Criticize! Criticism for Interpretability," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 2288–2296. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157096.3157352

[29] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR," *ArXiv: 1711.00399*, November 2017.

[30] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation," vol. 10, no. 7. Public Library of Science, 07 2015, pp. 1–46. [Online]. Available: https://doi.org/10.1371/journal.pone.0130140

[31] D. Shiebler, "Understanding Neural Networks with Layerwise Relevance Propagation and Deep Taylor Series," Available at http://danshiebler.com/2017-04-16-deep-taylor-lrp/ (consulted at November 5th 2018), April 2017.

[32] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.

[33] E. Kim, "Tensorflow tutorial for various Deep Neural Network visualization techniques," Available at https://github.com/1202kbs/Understanding-NN (consulted at November 6th 2018), Feb 2018.

[34] R. R. Selvaraju, "Yes, Deep Networks are great, but are they Trustworthy?" Available at https://ramprs.github.io/2017/01/21/Grad-CAM-Making-Off-the-Shelf-Deep-Models-Transparent-through-Visual-Explanations.html (consulted at November 5th 2018), Jan 2017.

[35] D. Bertsimas, J. Dunn, G. C. Velmahos, and H. M. A. Kaafarani, "Surgical Risk Is Not Linear: Derivation and Validation of a Novel, User-friendly, and Machine-learning-based Predictive Optimal Trees in Emergency Surgery Risk (POTTER) Calculator." *Annals of surgery*, vol. 268 4, pp. 574–583, 2018.

[36] D. Bertsimas, J. Dunn, C. Pawlowski, J. Silberholz, A. Weinstein, Y. Zhuo, E. Chen, and A. A. Elfiky, "Applied Informatics Decision Support Tool for Mortality Predictions in Patients With Cancer," *JCO Clinical Cancer Informatics*, pp. 1–11, 06 2018.

[37] I. AI, "Turning Data into Trusted Action," Available at https://www.interpretable.ai/solutions.html (consulted at November 1st 2018), 2018.

[38] DarkLight, "Darklight active defense expert system," Available online at https://www.darklight.ai/ (consulted at 5/11/2018), 2016.

[39] simMachines, "Predictions with the why," Available online at https://simmachines.com/ (consulted at 19/12/2018), 2004.

[40] D. Kahneman, *Thinking, fast and slow*. New York: Farrar, Straus and Giroux, 2011. [Online]. Available: https://www.amazon.de/Thinking-Fast-Slow-Daniel-Kahneman/dp/0374275637/ref=wl_it_dp_o_pdT1_nS_nC?ie=UTF8&colid=151193SNGKJT9&coliid=I3OCESLZCVDFL7

Distinguishing Different Types of Cancer with Deep Classification Networks

Mafalda Falcão Ferreira
Faculty of Engineering, University of Porto
Porto, Portugal
up201204016@fe.up.pt

Rui Camacho
Faculty of Engineering, University of Porto
Porto, Portugal
rcamacho@fe.up.pt

Luís Filipe Teixeira
Faculty of Engineering, University of Porto
Porto, Portugal
luisft@fe.up.pt

***Abstract*—Cancer is one of the most serious health problems of our time. One approach for automatically classifying tumor samples is to analyze derived molecular information. In this work, we aim to distinguish 3 different types of cancer: Thyroid, Skin, and Stomach. For that, we use the same methodology previously developed by Ferreira *et al.*, comparing the performance of a Denoising Autoencoder (AE), combined with two different approaches when training the classification model: (a) fixing the weights, after pre-training an AE, and (b) allowing fine-tuning of the entire network. We also apply two different strategies for embedding the AE into the classification network: (1) by only importing the encoding layers, and (2) by importing the complete AE. Our best result was the combination of unsupervised feature learning through a Denoising AE, followed by its complete import into the classification network, and subsequent fine-tuning through supervised training, achieving an F_1 score of 98.04% $\pm$ 1.09 when detecting thyroid cancer. Although there is still margin for improvement, we tend to conclude that the referred methodology can be applied to other datasets, generalizing its results.**

***Index Terms*—Cancer, Classification, Deep Learning, Autoencoders, Gene Expression Analysis.**

I. Introduction

Cancer is a generic term for a large group of diseases, which involve an abnormal cell growth with the potential of spreading to other parts of the body. In 2018, cancer was the second leading cause of death, worldwide. It was responsible for 9.6 million deaths, where approximately 70% occurred in developing countries [1].

Gene expression is the phenotypic manifestation of a gene or genes by the processes of genetic transcription and translation [2]. Its characterization can help to better understand cancer molecular basis, that can directly influence its prognosis, diagnosis, and treatment. Large scale cancer genomics projects, such as The Cancer Genome Atlas (TCGA - https://tcga-data.nci.nih.gov/) and the International Cancer Genome Consortium [3], try to translate gene expression, by cataloging and profiling through next-generation sequencing thousands of samples across different types of cancers. These projects are generating genome-wide gene expression assays datasets, having over 50k features representative of genes. Working with these datasets is somewhat challenging, due to (1) a small number of examples, (2) lack of balance distribution between classes, and (3) potential underlying noise, caused by eventual technical and biological covariates [4].

In this paper, we apply a previously developed methodology for gene expression datasets to the detection of 3 different types of cancer. Section II briefly presents an overview of recent related work in this area. Section III states our research approach and describes the chosen methodology, the data preprocessing step, the considered AEs for this study, and how we evaluate our results. In Section IV we present a discussion regarding the achieved results. Finally, Section V concludes this work, with an overview of the methodology, the results, and some future work considerations.

II. Related Work

The importance of correctly classify this disease is leading many research groups to experiment and study the application of Machine Learning algorithms, as an aim model the progression and treat cancerous conditions [5].

The authors in [6] used a predictive model based on the Multilayer Perceptron and Stacked Denoising Autoencoder (MLP-SAE), to assess how good genetic variants will contribute to gene expression changes. The MLP-SAE model is composed of 4 layers: one input, one output, and two hidden layers from two autoencoders (AEs). The model was trained using the Mean Squared Error (MSE) as the loss function. They first trained the AEs with a stochastic gradient descent algorithm to use them, *a posteriori*, on the multilayer perceptron training phase (*i.e.* weight initialization). Xie *et al.* used cross-validation to select the optimal model to (1) compared the performance of the proposed model with the Lasso and Random Forest methods, and (2) evaluate the performance of the model, when predicting the gene expression values, on an independent dataset. In (1) an MSE of 0.2890 of the MLP-SAE outperformed both previously referred methods (0.2912 and 0.2967, accordingly). In (2), the authors present a figure with the true expression *versus* the predicted expression while using the best MLP-SAE model and concluded that it can capture the changes in gene expression quantification.

Teixeira *et al.* [7] analyzed the combination of different methods of unsupervised feature learning — *viz.* Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA), Denoising Autoencoder (DAE), and

TABLE I: A sample of the thyroid dataset. The header line represents the names of the genes and column values represent its expression for each sample. *NA* means that a value is missing, for that gene, in a sample.

	sampleId	UBE2Q2P2_100134869	HMGB1P1_10357	LOC155060_155060	...	ZZEF1_23140	ZZZ3_26009	TPTEP1_387590	AKR1C6P_389932
0	TCGA-4C-A93U-01	-1.6687	NA	NA	...	-1.2094	-0.9478	-1.3739	NA
1	TCGA-BJ-A0YZ-01	-1.1437	NA	NA	...	0.9139	-0.4673	-0.0166	NA
2	TCGA-BJ-A0Z0-01	-0.9194	NA	NA	...	1.3579	2.1918	-1.5856	NA
3	TCGA-BJ-A0Z2-01	1.1382	NA	NA	...	0.2969	1.5512	-1.5897	NA
4	TCGA-BJ-A0Z3-01	-0.3333	NA	NA	...	-0.1101	0.4926	-1.3379	NA
5	TCGA-BJ-A0Z5-01	-0.1493	NA	NA	...	-0.3399	0.0004	-0.6436	NA
6	TCGA-BJ-A0Z9-01	-0.3003	NA	NA	...	0.7398	0.5328	0.7169	NA
7	TCGA-BJ-A0ZA-01	-1.2039	NA	NA	...	1.7924	0.7504	-0.2503	NA
8	TCGA-BJ-A0ZB-01	-0.3300	NA	NA	...	0.6452	0.8508	-0.5903	NA
9	TCGA-BJ-A0ZC-01	0.6342	NA	NA	...	1.3777	0.6752	1.4248	NA

Stacked Denoising Autoencoder — with different sampling methods for classification purposes. They studied the influence of the input nodes on the reconstructed output of the AEs, when feeding these combinations results to a *shallow* artificial network, for the classification task of Papillary Thyroid Carcinoma. The combination that achieved the best results was a SMOTE and Tomek links, with a KPCA, with a mean F_1 score of 98.12% in 5-fold cross-validation. The authors, however, preferred the usage of a DAE, for which they affirm yielded similar results (though with a mean F_1 score of 94.83%). The size of the encoded representation was chosen to be slightly smaller than the dataset size (400).

In [8], the authors developed a methodology for the detection of papillary thyroid carcinoma. Ferreira *et al.* studied and compared the performance of a deep neural network classifier architecture, where they used autoencoders (AEs) as a weight initialization method. The AEs were pre-trained to minimize the reconstruction error and subsequently used to initialize the top layers weights of the classification network, with two different strategies: (1) Just the encoding layers, and (2) All the pre-trained AE. 6 types of AEs were used: Basic AE, Denoising AE, Sparse AE, Denoising Sparse AE, Deep AE, and Deep Sparse Denoising AE. Sampling, data augmentation, and normalization techniques when pre-processing the data were absent from their approach. To evaluate and support the results, the authors used stratified 5-fold cross-validation to split the data into training and validation partitions, providing 4 different metrics: Loss, Precision, Recall, and F_1 score. Their best result was the combination of unsupervised feature learning through a single-layer Denoising AE, followed by its complete import into the classification network, and subsequent fine-tuning through supervised training, achieving an F_1 score of 99.61%, with a variance of 0.54.

III. Methodology

In this work, we use the methodology previously developed by Ferreira *et al.* [8] to detect 3 types of cancer, instead of distinguishing cancerous from healthy samples. We analyze the unsupervised learning loss functions evolution separately during their training phase, for all the AEs. We chose to only compare the performance of the AE that had the best result in [8] (Denoising AE) as weight initialization of a classification architecture, studying two different approaches for weight initialization and two different strategies for embedding the AE layers. In the end, we discuss the 5-fold cross-validation results with its variances to conclude how similar or how dissimilar are our results compared to the ones previously reached.

We used 3 different RNA-Seq datasets from The Cancer Genome Atlas (TCGA), which one representing a type of cancer: thyroid, skin, and stomach. A sample of the data is presented in Table I. All three datasets are composed of the same 20442 features. Each feature represents a certain gene, where its values represent the expression of that gene, for a certain sample. The thyroid cancer dataset has 509 examples, the skin cancer dataset 472 and the stomach cancer dataset 415. Further pre-processing of data is described in Section III-B.

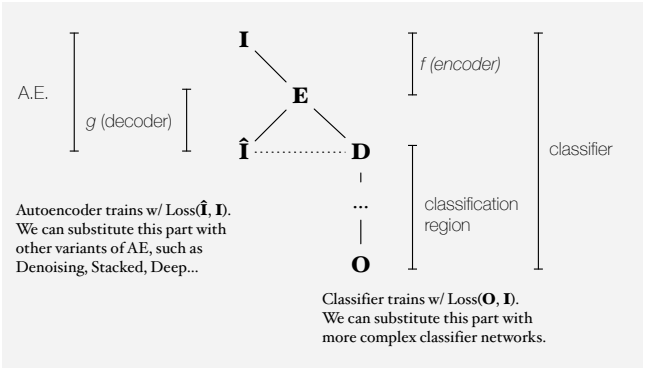
A. Overall Description

As in [8], our experiment consists in the performance comparison of a deep neural network classifier architecture, where we vary its top layers. We pre-train the autoencoders to minimize the reconstruction error and subsequently use them to initialize the top layers weights of the classification network, with two different strategies: **(1)** Just the encoding layers, and **(2)** All the pre-trained autoencoder.

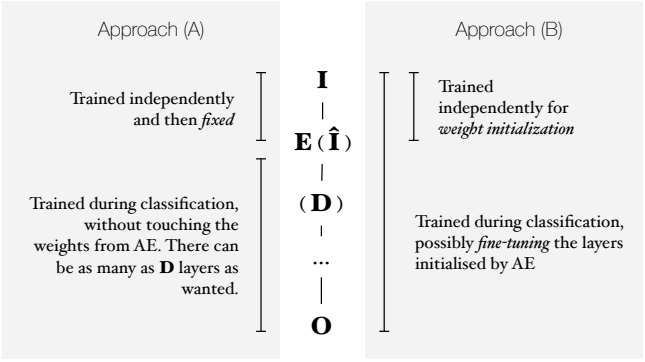
Each architecture is thus trained to classify the input data as either *Thyroid*, *Skin* or *Stomach*, accordingly to the type of cancer. We use the same architecture as Ferreira *et. al* and adapted this multi-label classification problem to a binary classification one: for a type of cancer C, we train the model to detect C and not C, instead of detecting cancer and healthy samples. Besides the top layers imported from the AE, the classification region of the full network starts with a Batch Normalization layer [9], and proceeds with two Fully Connected layers using Rectified Linear Unit (ReLU) activation [10]; the last one — prediction layer — is a single neuron layer with a Sigmoid non-linearity [11].

B. Data Pre-processing

We pre-process the datasets separately. We start removing, for each dataset, the features that had the same value for all the instances in the dataset. When a value is constant for all the examples, there is no entropic value (*i.e.*, it is not possible to infer any information). We then imputed the remain missing values with the average value of its respective column, and added (to each one) a column *Label* to match each instance to its type of cancer. Our goal is to distinguish different from



(a)



(b)

Fig. 1: (a) General overview and (b) two approaches under study. I represents the input layer, E the encoding layer(s), $\hat{I}$ the reconstructed input of the autoencoder, D the dense layers (from the classification region), and O the classifier's output. Note that, in some cases, $\hat{I}$ might be a better starting point than E, as we will observe in the discussion.

cancer, so we assign a positive value (1) to the class we want to predict, and 0 to the remaining ones. When training the model to detect:

- **Thyroid cancer**: All thyroid examples are labeled as 1 and the skin and stomach instances as 0.
- **Skin cancer**: All skin examples are labeled as 1 and the thyroid and stomach instances as 0.
- **Stomach cancer**: All stomach examples are labeled as 1 and the thyroid and skin instances as 0.

However, after this process, it is not guaranteed (and actually quite unlikely) that the same features will be removed in the 3 cancer datasets. Thus, when merging the 3 sets of data, we only use their intersection, so that the different types of cancer are represented by the same features. After the full data pre-processing, the final dataset has 18321 feature columns and 1396 examples (36% of thyroid cancer, 34% of skin cancer, and 30% of stomach cancer).

C. Autoencoders

Each experience consists in (a) training a different type of AE, and then (b) using it to initialize the top layers of the classification architecture, *viz.* all the encoding layers and all the AE (see Figure 1). Each of these new representations is subsequently attached to the classification architecture, thus fulfilling Strategies 1 and 2, mentioned in Section III-A.

An AE is a neural network, trained for the output to be the same as the input. One can obtain useful features from an AE by constraining a hidden layer to have smaller dimensionality than the original input, in which cases they are named *undercomplete* [12]. Such constraint forces the AE to capture the most salient features of the training data. Let f and g correspond to the encoding and decoding functions of the AE, parameterized on θ_e and θ_d respectively, where $\theta = \theta_e \cup \theta_d$, L being an appropriate loss function, and J the cost function to be minimized[1]. The AE learning process attempts to find a value for θ that minimizes:

$$\underset{\theta}{\operatorname{argmin}}\ J(\theta, X) = L(X, g_{\theta_d}(f_{\theta_e}(X)) \tag{1}$$

Penalizing a reconstruction of the input $\hat{X} = g_{\theta_d}(f_{\theta_e}(X))$ from being dissimilar from the original data X. In this work, we use MSE as the loss function:

$$L(X, \hat{X}) = \sum (X - \hat{X})^2 \tag{2}$$

and ReLU as the activation function for all layers of the AEs. We compare the performance of 6 different AEs as an initialization method [13], *i.e.* first we pre-train an AE, and then we import part or all of its layer(s) to the classifier architecture, as shown in the Figure 1.

1) Basic (one-layer) Autoencoder (AE): The most basic AE is composed of a single hidden layer [12]. In this work, we fix its size at 128. This AE learning process follows the Equation 1, presented in Section III-C. When combining linear activations and MSE loss functions, *undercomplete* AEs learn to span the same subspace as Principal Component Analysis (PCA) [14].

2) Denoising Autoencoder (DAE): The DAE [15] is an autoencoder that tries both to encode the input and preserve its information to undo the effect of a corruption process applied to the input of the AE, by:

$$\underset{\theta}{\operatorname{argmin}}\ J(\theta, X) = L(X, g_{\theta_d}(f_{\theta_e}(\tilde{X}))) \tag{3}$$

where $\tilde{X}$ is a copy of the input X, corrupted by some form of *noise* [14]. In our case, we apply a *Dropout* layer, directly after the input layer as a form of Bernoulli Noise [16], where we randomly delete 10% of the connections. The hidden encoding layer size is 128.

[1]Some authors use *cost* and *loss* function interchangeably. For the purpose of this paper, the cost, or objective function (J) includes the proper loss function (L) and possibly other components such as regularization. Whenever reporting loss, we always refer to (J).

TABLE II: Performance comparison of the classifier. We are only importing the top layers from a DAE since it was the AE that led to better results in [8]. *T* represents thyroid cancer detection, *Sk* skin cancer detection, and *St* stomach cancer detection. When measuring loss, lower is better. For all the other metrics, higher is better. All the values presented are the 5-fold x-validation average, at the validation set, by selecting the best performing model according to its F_1 *score*.

	Fixed Weights (Approach A)					Fine-Tuning (Approach B)				
Top Layers (DAE)	Loss	Accuracy (%)	Precision (%)	Recall (%)	F_1 score (%)	Loss	Accuracy (%)	Precision (%)	Recall (%)	F_1 score (%)
T: Encoding Layers	0.309 ± 0.37	89.12% ± 2.44	82.80% ± 4.36	88.81% ± 3.69	**85.63% ± 3.08**	0.117 ± 0.50	98.35% ± 0.86	96.06% ± 2.13	99.61% ± 0.54	97.79% ± 1.13
T: Complete Autoencoder	0.375 ± 0.16	92.41% ± 2.69	86.82% ± 6.09	93.91% ± 1.28	90.11% ± 3.09	0.662 ± 0.49	98.57% ± 0.80	97.88% ± 1.83	98.23% ± 1.76	**98.04% ± 1.09**
Sk: Encoding Layers	0.346 ± 0.46	88.82% ± 2.36	91.19% ± 3.75	74.34% ± 8.33	81.60% ± 4.91	0.545 ± 0.02	98.57% ± 1.13	**100.00% ± 0.00**	95.76% ± 3.37	97.81% ± 1.76
Sk: Complete Autoencoder	0.482 ± 0.07	91.33% ± 2.55	88.02% ± 6.69	86.65% ± 4.03	87.16% ± 3.53	0.893 ± 0.03	98.14% ± 0.46	98.27% ± 0.59	96.19% ± 0.94	97.22% ± 0.70
St: Encoding Layers	0.431 ± 0.06	83.16% ± 4.14	90.70% ± 5.96	47.95% ± 13.32	62.01% ± 12.99	0.590 ± 0.03	98.57% ± 0.72	99.25% ± 0.68	95.90% ± 2.19	97.54% ± 1.25
St: Complete Autoencoder	0.465 ± 0.12	89.83% ± 2.18	85.13% ± 3.14	80.00% ± 9.36	82.16% ± 4.90	0.147 ± 0.10	97.49% ± 1.81	97.95% ± 1.50	93.49% ± 5.15	95.63% ± 3.23

3) Sparse Autoencoder (SAE): A SAE is an AE whose learning process involves not just the minimization of the reconstruction error, but also a *sparsity penalty* (Ω) applied to the encoding parameters θ_e:

$$\underset{\theta}{\text{argmin}}\ J(\theta, X) = L(X, g_{\theta_d}(f_{\theta_e}(X))) + \lambda \cdot \Omega(\theta_e) \qquad (4)$$

An AE that has been regularized to be sparse must respond to unique statistical features of the dataset it has been trained on, rather than simply reproducing X [14] [17]. Here we use an L1 penalty with a λ of 10^{-5} as the sparsity component, and an encoding size of 128.

4) Denoising Sparse Autoencoder (DSAE): The DSAE has the characteristics of both SAE and DAE:

$$\underset{\theta}{\text{argmin}}\ J(\theta, X) = L(X, g_{\theta_d}(f_{\theta_e}(\tilde{X}))) + \lambda \cdot \Omega(\theta_e) \qquad (5)$$

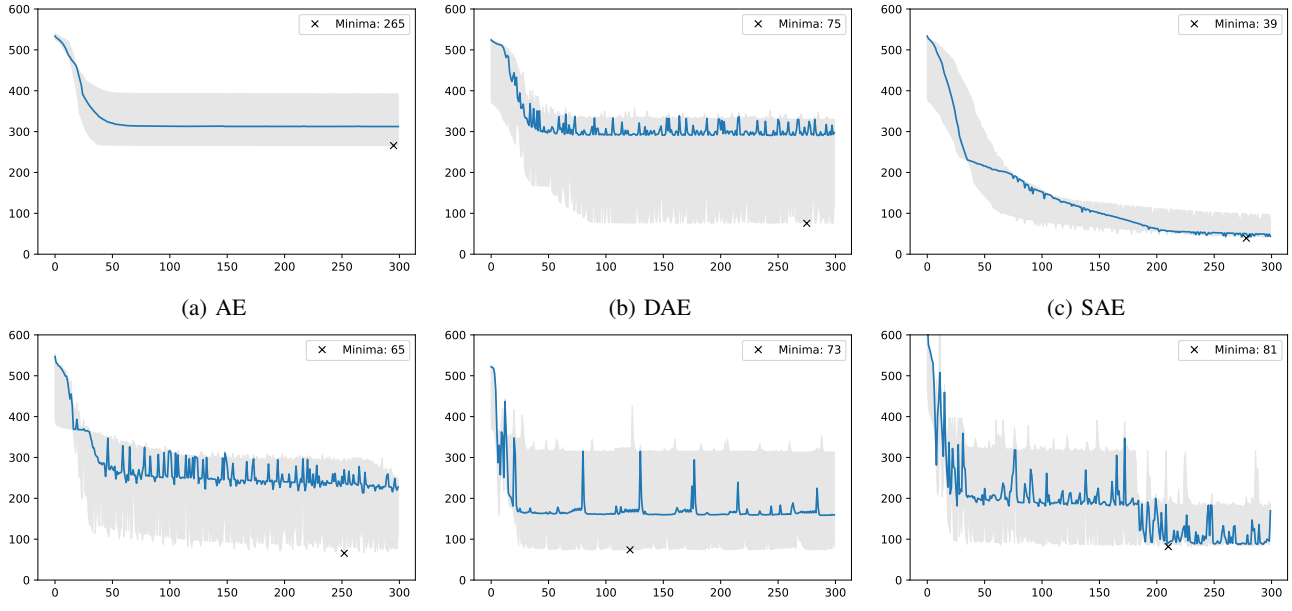
DSAEs try to encode the input data, reconstruct it without noise (10% *Dropout* at the encoding layer, with size 256), and minimize the output error, with an L1 penalty and a λ of 10^{-5} in the hidden layer.

5) Deep Autoencoder (DpAE): A DpAE is similar to the Basic AE. The difference between the two types is the number of hidden layers. We built a DpAE with 3 encoding layers in decreasing size (512, 256, and 128). Its learning process focuses on reducing the error of the input reconstruction, exactly as the Basic (one-layer) AE (Equation 1).

6) Deep, Sparse, Denoising Autoencoder (DpSDAE): A DpSDAE is a combination of above different types of AEs, with the loss function as in Section III-C4. The DpSDAE attempts to (1) encode the input data, (2) remove the input noise caused by a 10% *Dropout* during encoding, and (3) minimize, during training, the output error. The hyperparameters are the same as given above, and its loss function is the same presented in Equation 4.

D. Evaluation

We use stratified 5-fold cross-validation to split the data into training and validation partitions, to ensure statistical



(a) AE (b) DAE (c) SAE

(d) DSAE (e) DpAE (f) DSDAE

Fig. 2: Loss values on the training set for the 300 epochs of autoencoder training, with corresponding minimas.

significance. Each AE and classifier are trained during 100 and 300 epochs, respectively, with a batch size of 500. The loss of the classifier model is calculated by the *categorical cross-entropy* [14], and trained using an *adam* optimizer [18]. We proceed to evaluate its performance through 4 additional metrics: Accuracy, Precision, Recall, and F_1 scorealso for the training and the validation sets.

IV. Results and Discussion

One tends to assume that the previously described methodology can be generalized to other datasets and problems: Importing the complete pre-trained DAE to the upper layers of the classification architecture and allowing subsequent *fine-tuning* achieved the best overall performance, with an F_1 score of 98.04% (when detecting thyroid cancer), a result that is quite close to the overall best reported in [8]. However, for both detection of skin and stomach cancers, the best-achieved result was, respectively, 97.81% ($\pm$ 1.76) and 97.54% ($\pm$ 1.25), where the combination differs only on the DAE layers that are embedded into the classifier (only the encoding layers). We may assume that this methodology can generalize to other types of data.

Fine-tuning (Approach B) leads to better results than fixing the weights (Approach A): In [8], the authors claimed that their results cannot support that Approach B gave better results than Approach A. However, with our data, it is clear that fine-tuning the weights of the top layers leads to better results, by a margin of 10 – 20%, when considering the F_1 score metric, as one can see in Table II.

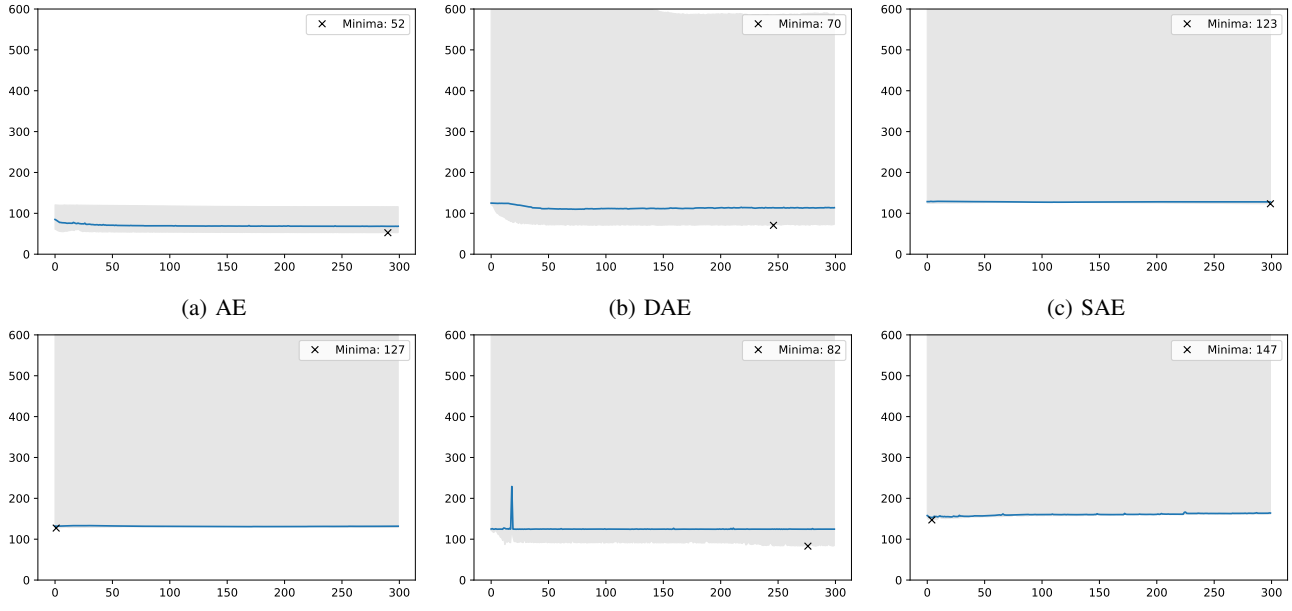
There is not enough evidence to support the assumption that the overall usage of AEs seem to capture the most relevant information for the task: Although our overall best was close to the overall best of the previously referred work, there is a big difference between the two approaches of weight initialization when experimenting our data. Also, there is a big divergence when analyzing the AEs curves in the train and validation phases, as it is observable in Figure 3 and Figure 2. One may assume that the AEs learning process is being compromised given that (1) in some cases, in the validation phase (for example the DSAE – Figure 3d – and the DSDAE – Figure 3f), the minima is found too early and (2) the data split in the cross-validation may have influence on the learning process.

V. Conclusions

In this work, we compared the performance of different autoencoders (AEs) as an unsupervised initialization method for deep classification neural networks. For that, we used the methodology described in [8]: we combined a DAE with two different approaches when training the classification architecture: (a) by fixing the imported weights, and (b) by allowing them to be fine-tuned during supervised training. We tried two different strategies for embedding the DAE into the classification network: (1) using the encoding layers as weight initialization, and (2) using the complete AE, *i.e.*, both the encoding and decoding layers.

During this work, we faced some obstacles during the preparation of the data, and while training the AEs and the classification network:

1) **Dealing with missing values is not an easy task**: deciding *"what strategy to use?"* when imputing is, most of the times, a guess, and its impact on the training phase is always unknown.



(a) AE (b) DAE (c) SAE

(d) DSAE (e) DpAE (f) DSDAE

Fig. 3: Loss values on the validation set for the 300 epochs of autoencoder training, with corresponding minimas.

2) **Joining datasets is not easy, as well**: one hopes that, when merging different datasets, replacing the undefined values reasonable. However, that process leads to a leak of information when there are columns that only one of the labels has or, which is highly discriminatory. Let's suppose that we have two datasets: one from the workers of organization A, and another one representative of the workers of organization B. The organization B dataset has the height of the workers and the organization A dataset does not have that feature. The network will easily detect who are the workers from organization A - the ones that do not have value in the height feature. Even if we filled the missing values in the organization A dataset, the exact imputed values would be easily associated by the network to the organization A workers. In these cases, the network will only *memorize* the examples instead of *learning* from them.
3) **It is important to analyze the evolution of the training phase curve, but it is even more important to analyze the evolution of the validation phase curve**: the learning behavior can be drastically different in both phases, as observed in Figure 3 and Figure 2, and they both should be studied not just to detect *overfitting*, but also to make other assumptions, such as *"Maybe there is something wrong with my data!"*.
4) **The tools for the analysis of the results are easier to use on a binary classification scenario**: frequently, evaluation metrics (such as Precision, Recall, and F_1 score) need to know what is the label of interest, and one can only give one value in the case of binary classification.

With this, one can say that the process of Data Science is long, tedious, full of *trial and error*, and difficult when there is no *"human"* description associated to the data we are dealing with.

If we establish as baseline the results of Ferreira *et al.*, we can assume that it may be possible to generalize this process to other datasets and problems. Importing a complete pre-trained DAE to the top layers of the classifier (Strategy 2), followed by fine-tuning (Approach B), when detecting Thyroid cancer, achieved the best overall results, with an F_1 score of 98.04% with a variance of 1.09. Fine-tune led to better results, boosting the results between 10 and 20% in the F_1 score metric. Contrary to the results obtained in the mentioned previous work, there is not enough evidence to support the assumption that the overall usage of AEs seems to capture the most relevant information for the task. Finally, we assume that our results can still be improved, by (1) further investigation on the best strategies when preparing the data, (2) increasing the models training time, and (3) integrating other types of AE in the classifier architecture.

REFERENCES

[1] WHO, "Cancer - fact sheet," https://www.who.int/en/news-room/fact-sheets/detail/cancer.

[2] "Gene expression - ncbi - nih," https://www.ncbi.nlm.nih.gov/probe/docs/applexpression/.

[3] T. I. C. G. Consortium and e. a. Hudson (Chairperson), "International network of cancer genome projects," *Nature*, vol. 464, pp. 993 EP –, Apr 2010, perspective.

[4] K. R. Kukurba and S. B. Montgomery, "Rna sequencing and analysis," *Cold Spring Harb Protoc*, vol. 2015, no. 11, pp. 951–969, Nov 2015.

[5] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and Structural Biotechnology Journal*, vol. 13, pp. 8 – 17, 2015.

[6] R. Xie, J. Wen, A. Quitadamo, J. Cheng, and X. Shi, "A deep auto-encoder model for gene expression prediction," *BMC Genomics*, vol. 18, no. 9, p. 845, Nov 2017.

[7] V. Teixeira, R. Camacho, and P. G. Ferreira, "Learning influential genes on cancer gene expression data with stacked denoising autoencoders," *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1201–1205, 2017.

[8] M. F. Ferreira, R. Camacho, and L. F. Teixeira, "Autoencoders as weight initialization of deep classification networks applied to papillary thyroid carcinoma," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec 2018, pp. 629–632.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, pp. 448–456.

[10] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. USA: Omnipress, 2010, pp. 807–814.

[11] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, Dec 1989.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362.

[13] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 153–160.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[15] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1096–1103.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[17] A. Ng, "Cs294a lecture notes - sparse autoencoder." [Online]. Available: https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

Reinforcement Learning to Reach Equilibrium Flow on Roads in Transportation System

Hajar Baghcheband
Computer Engineering
University of Porto
Porto, Portugal
h.baghcheband@fe.up.pt

***Abstract*—Traffic congestion threats the vitality of cities and the welfare of citizens. Transportation systems are using various technologies to allow users to adapt and have a different decision on transportation modes. Modification and improvement of these systems affect commuters' perspective and social welfare. In this study, the effect of equilibrium road flow on commuters' utilities with a different type of transportation mode will be discussed. A simple network with two modes of transportation will be illustrated to test the efficiency of minority game and reinforcement learning in commuters' daily trip decision making based on time and mode. The artificial society of agents is simulated to analyze the results.**

***Index Terms*—Transportation system, minority game, reinforcement learning, multi-agent system, simulation**

I. Introduction

Transport is an activity where something is moved between the source and destination by one or several modes of transport. There are five basic modes of transportation: road, rail, air, water and pipeline [1]. An excellent transport system is vital for a high quality of life, making places accessible and bringing people and goods together. Information and Communication Technologies (ICT) helps to achieve this high-level objective by enhancing Transportation systems with intelligent systems[2].

Among all type of transport, road transport has an important role in daily trips. People are migrated to cities because of the benefits of services and employment compared to rural areas. However, it is becoming harder to maintain road traffic in smooth working order. Traffic congestion is a sign of a city's vitality and policy measures like punishments or rewards often fail to create a long term remedy. The rise of ICT enables the provision of travel information through advanced traveler information systems (ATIS). Current ATIS based on shortest path routing might expedite traffic to converge towards the suboptimal User Equilibrium (UE) state[3].

Traffic congestion is one of the reasons for negative externalities, such as air pollution, time losses, noise, and decreasing safety. As more people are attracted to cities, future traffic congestion levels are not only decreased but also increased and extending road capacity would not solve congestion problems. While private cars maximize personal mobility and comfort, various strategies have attempted to discourage car travel to use public transportation.

To encourage commuters to shift from private car to public transport or intermodal changes, it is exigent to provide a competitive quality to public transport compared to its private counterpart. This can be measured in different aspects such as safety, comfort, information, and monetary cost, but more importantly, travel times compared to those of private cars[4].

Moreover, Policy measures in transportation planning aim at improving the system as a whole. Changes to the system that result in an unequal distribution of the overall welfare gain are, however, hard to implement in democratically organized societies [5]. Different categories of policies can be considered in urban road transportation: negative incentives [6], positive incentives or rewards[7], [8], sharing economy [3], [9].

In recent researches, positive policies were discussed as discount or money payback to commuters. Kokkinogenis et al. [12], discussed a social-oriented modeling and simulation framework for Artificial Transportation Systems, which accounts for different social dimensions of the system in the assessment and application of policy procedures. They illustrated how a social agent-based model can be a useful tool to test the appropriateness and efficiency of transportation policies[12].

Traditional transport planning tools are not able to provide welfare analysis. In order to bridge this gap, multi-agent microsimulations can be used. Large-scale multi-agent traffic simulations are capable of simulating the complete day-plans of several millions of individuals (agents) [10]. A realistic visualization of agent-based traffic modeling allows creating visually realistic reconstructions of modern or historical road traffic. Furthermore, the development of a complex interactive environment can bring scientists to new horizons in transport modeling by an interactive combination of a traffic simulation (change traffic conditions or create emergencies on the road) and visual analysis[11].

Klein et al. developed a multi-agent simulation model for the daily evolution of traffic on the road that the behavior of agents was reinforced by their previous experiences. They considered various network designs, information recommendations, and incentive mechanisms, and evaluated their models based on efficiency, stability and equity criteria. Their results concluded that punishment or rewards were useful incentives[3].

To improve the behavior of agents, reinforcement learning

is one of the key means in multi-agent systems. reinforcement learning techniques recently proposed for transportation applications and they have demonstrated impressive results in game playing. Nallur et al. introduced the mechanism of algorithm diversity for nudging system to reach distributive justice in a decentralized manner. They use minority game as an exemplar of an artificial transportation network and their result showed how algorithm diversity lead to faired reward distribution[19].

The main goal of this study is to develop a model, based on the concept of minority games and reinforcement learning, to achieve equilibrium flow through public and private transportation. Minority game is applied to consider rewards, positive policy, for winner and learning is a tool to increase the user utility based on rewards. To illustrate, an artificial society of commuters are considered instantiated on a simple network with two modes of transportations, namely public (PT) and private (PR).

The remaining parts are organized as follow. In Section II, the conceptual framework will be discussed and consists of a definition of user utility, minority game, and reinforcement learning algorithms. Illustration scenario of network and commuters and initial setup are explained in Section III. Experiments and results are shown in Section IV. Conclusion of the hypothesis and results are drawn in Section V.

II. Description of the Framework

In this section, the theoretical aspects and methodological ones are described, and also network design and model will be discussed.

A. Traffic Simulation

Traffic simulation models are classified into macroscopic and microscopic models. The hydrodynamic approach to model traffic flow is typical for macroscopic modeling. With this kind of approach, one can only make statements about the global qualities of traffic flow. For observing the behavior of an individual vehicle a microscopic simulation is necessary. Because traffic cannot be seen as a purely mechanical system, a microscopic traffic simulation should also take into consideration the capabilities of human drivers (e.g., perception, intention, driving attitudes, etc.)[2].

B. Network Design

The network is formally represented as graph G (V, L) which V is the set of nodes such as *Origin*, *Destination,* and *middle* nodes and L is the set of roads (edges or links) between nodes[3], [12]. Each line $l_k \in$ L has some properties such as mode, length, and capacity. In addition, the volume-delay function is used to describe the congestion effects macroscopically, that is, how the exceeding capacity of flow in a link affects the time and speed of travel, as below [13]:

$$t_k = t_{0k}\ [1+\alpha\ (X_k/C_k)^{\beta}] \tag{1}$$

where t_{0k} is free flow travel time, X_k is the number of vehicle and C_k shows the capacity of the link k, α and β are controlling parameters.

C. Commuters Society

Commuters, agents of the artificial society, have some attributes regarding travel preferences such as time (desired arrival time, desired travel time, mode of transportation, mode flexibility), cost (public transportation fare, waiting time cost, car cost if they have), socioeconomic features (income).

They will learn and make a decision for their dairy plan based on their daily expectation and experience. The iteration module generates the demands of the transportation modes and desired time. Daily trips schedule for a given period of the day and define the set of origins and destinations with the respective desired departure and arrival times to and from each node.

The utility-based approach is considered to evaluate travel experience and help agents make decisions. Total utility is computed as the sum of individual contribution as follow and is the combination based on previous researches [5], [12]:

$$\mathrm{U_{total}} = \sum_{i=1}^{n} \mathrm{U_{perf,\ i}} + \sum_{i=1}^{n} \mathrm{U_{time,\ i}} + \sum_{i=1}^{n} \mathrm{U_{cost,\ i}} \tag{2}$$

where $\mathrm{U_{total}}$ is the total utility for a given plan; n is the number of activities, which equals the number of trips (the first and the last activity are counted as one); $\mathrm{U_{Pref,\ i}}$ is the utility earned for performing activity i; $\mathrm{U_{time,\ i}}$ is the (negative) utility earned by the time such as travel time and waiting time for activity i; and $\mathrm{U_{cost,\ i}}$ is the (usually negative) utility earned for traveling during trip i.

1) Performance Utility : To measure the utility of selecting activity *i,* each mode of transportation has different variables. For public mode, comfort level and bus capacity, and for private, pollution and comfort level are considered.

2) Time Utility: The measurement of the travel time quantifies the commuter's perception of time based on various components like waiting and in-vehicle traveling. Waiting time indicates the service frequency of public transportation. In-vehicle traveling time is an effective time to travel from origin to destination.

3) Monetary Cost Utility: Monetary cost can be defined as fare cost of public transportation, cost of fuel, tolls (if exists), car insurance, tax, and car maintenance. This kind of cost will be measured based on the income of commuters.

Regard to three different utilities, the total utility of public and private can be measured as follow:

$$\mathrm{U^{total}_{private}} = \sum_{i=1}^{N} \mathrm{U^{i}_{private}} \tag{3}$$

$$\mathrm{U^{i}_{private}} = \left(\alpha_{\mathrm{time}} * \left(\mathrm{t^{i}_{tt,exp}}/\mathrm{t^{i}_{tt}}\right)\right) + \left(\beta_{\mathrm{PR}} * \left(\mathrm{cost\_PR}/\mathrm{income_i}\right)\right) + \left(\alpha_{\mathrm{pollution}} * \mathrm{t^{i}_{tt}} * \mathrm{pollution}\right) + \alpha_{\mathrm{com\_PR}} * \left(\mathrm{t^{i}_{tt,exp}}/\mathrm{t^{i}_{tt}}\right) \tag{4}$$

$$\mathrm{U^{total}_{public}} = \sum_{i=1}^{N} \mathrm{U^{i}_{public}} \tag{5}$$

$$U^{i}_{\text{public}} = \left(\alpha_{\text{time}} * (t^{i}_{\text{tt,exp}} / t^{i}_{\text{tt}})\right) + (\beta_{\text{PT}} * (\text{cost\_PT}/\text{income}_{i})) + \alpha_{\text{com\_PT}} * \left(t^{i}_{\text{wt,exp}} / t^{i}_{\text{wt}}\right) + \left(\alpha_{\text{cap}} * t^{i}_{\text{tt}} * \text{capacity}^{\,i}_{\text{exp}} / \text{bus\_capacity}\right) \quad (6)$$

where t^{i}_{tt} and $t^{i}_{tt,exp}$ are total travel time and expected total travel time of agent *i*, cost_PR is the monetary cost of private transportation (fuel, car maintenance and etc.), cost_PT, the fare of public transportation, $income_i$, the agent's income per day, pollution is the amount of pollution is produced by private vehicles, capacity $^{i}_{exp}$, and bus_capacity are expected capacity of the bus and the total capacity of each bus respectively, $t^{i}_{wt,exp}$, expected waiting time and t^{i}_{wt} is the waiting time by agent *i*. α_{time}, β_{PT}, β_{PR}, $\alpha_{pollution}$, α_{com_PR}, α_{com_PT} and α_{cap} are considered as marginal utilities or preferences for different components.

D. Minority Game

The minority game, introduced by Challet and Zhang (1997)[14], consisting of *N* agents (*N* is an odd number). They have to choose one of two sides independently and those on the minority side win. Winner agents get reward points, nothing for others. Each agent draws randomly one out of his *S* strategies and uses it to predict the next step. To choose what strategy to use each round, each is assigned a score based on how well it has performed so far, the one with the leading score is used at a time step.

It was originally developed as a model for financial markets, although it has been applied in different fields, like genetics and transportation problems[15]. While simple in its conception and implementation, it has been applied in various fields of transportation such as public transportation[16], route choosing[17], road user charging scheme[18]. It can be useful in traffic management which travelers try to find less crowded and congestion roads.

E. Reinforcement Learning Method

Reinforcement learning (RL) is a class of machine learning concerned with how agents ought to take actions in an environment so as to maximize cumulative reward [19]. Alvin Roth and Ido Erev developed a new algorithm, which is called "Roth-Erve"[20], to model how humans perform in competitive games against multiple strategic players. The algorithm specifies initial propensities (q_0) for each of *N* actions and based on reward (r_k) for action (a_k) the propensities at the time ($t+1$) are defined as[20]:

$$q_j(t+1) = (1-\phi)\, q_j(t) + E_j(\in, N, k, t) \quad (7)$$

$$E_j(\in, N, k, t) = \begin{cases} r_k(t)[1-\in] & \text{if } j = k \\ r_k(t) * (\in/N-1) & \text{otherwise} \end{cases} \quad (8)$$

Where ϕ is recency as forgetting parameter and is exploration parameter. The probability of choosing action *j* at time *t* is:

$$P_j(t) = q_j(t) / \sum_{n=1}^{N} [q_n(t)] \quad (9)$$

III. Illustrative Scenario

In the simulation step, the perspective of the conceptual framework was considered a simple scenario where commuters make a decision over transportation mode and time during morning high-demand peak hour. Simulation model implemented through NetLogo [21] agent-based simulation environment.

A. Network and Commuters

In this study, two different links of two modes (PT or PR) consist of two middle nodes on each link. As it is shown in Fig. 1, to simplify the upper link is for private and the other for public transportation where each road is composed of one-way links.

Commuters, as type of agents, is defined by a number of state variables which are: (1) desired departure and arrival times, (2) experienced travel time, (3) the uncertainty they experienced during the trip with a given transportation mode, (4) a set of preferences about the transportation mode, (5) the perceived comfort as personal satisfaction for the mode choice, and (6) a daily income variable. While the agent experiences its travel activities, the costs associated with the different transportation mode, the perceived satisfaction of traveling (expressed in terms of travel times and comfort) and rewards earned by winners will have a certain impact on its mode and time choices.

Commuters can choose between traveling by PT or PR modes based on own-car value. The decision-making process of each agent is assumed to maximize the utility and flow equilibrium on roads. They perceive current traffic condition as well as previous experience and use this information in making a decision.

At the end of the travel each commuter stores the experienced travel time, costs, and crowding level (for PT mode users only) and emissions. These variables will be used to calculate the following day's utility. After that each agent evaluates its own experience, comparing the expected utility to the effective utility.

Based on minority game, we considered the number of commuters on each road and type of transportation and regard to Roth-Erve learning, the reward assigned to the winner who is in minority number and has the following criteria:

1) *Their obtained utility ($U_{effective}$) is greater than the utility prediction ($U_{expected}$) as below:*

$$U_{\text{effective}} > \alpha * U_{\text{expected}} \quad (10)$$

α is marginal preference.

2) *The obtained utility of agent is higher than mean utility in the whole network:*

$$U_{effective} \quad U_N \qquad \text{where} \quad U_N = \frac{1}{N} \sum_{i=1}^{N} U^i_{effective} \qquad (11)$$

Based on reward, effective utility they earned in their daily trip, car-ownership and mode-flexibility, each commuter decides about their new mode and time.

B. Initial Setup

As it is written in TABLE I, The capacity for all links was considered 150 and max capacity for each bus was 70 people. Population consist of 201 commuters was created, the odd number to coordinate with minority game, and they iterated their daily trips in 60 days. They were characterized by a number of attributes such as departure and arrival times, mode, daily income, car-ownership, and flexibility. Car-ownership is a Boolean variable and indicates if the agent is a private or public transportation user. Flexibility reflects the willingness of a private mode user to change its mode.

All agent's plans were done in rush hours of the day from 6:30 a.m. to 10:30 a.m., with a normal distribution to simulate peak times. It was observed a high demand in peak duration between 8- 9:30 a.m., on both roads. The range of income was 20 to 70 Euro per day. The routes between nodes *Origin* and *Destination* had both a length of 19 km.

The free-flow travel time from node *Origin* to *Destination* was approximately 25 minutes in the PR mode and for the public transportation, around 35 minutes plus the waiting time at the bus stop and walking time. The bus frequency service was 10 minutes before the rush hour and 5 minutes during the rush hour.

IV. Results and Experiments

We performed sixty iterations of the model, Roth-Erve learning was used to establish the equilibrium commuters between both roads along the departure time interval. During simulation steps, we monitored agents' expected and effective utilities, average travel times of public and private transportation, average total travel times, number of commuters of each mode and differences between the average of total travel time in public and private transportation.

Table I
Default Value Of Network And Learning Parameters

Variable	Value
Number of commuters	N=201
Capacity of links	L_i= *150*
Capacity of bus	B=70
Time	6:30 a.m. to 10:30 a.m.
Range of income	20 to 70 per day
Simulation period	60 days
Recency (φ)	0.3
Exploration (ε)	0.6

The propensity of commuters to select public and private were set by normal distribution random and updated based on recency and exploration learning parameters. Earned scores and two propensities were observed during all days.

In Fig. 2, the distribution of all commuters is depicted among all days, where green line and the red line show the number of agents on roads with public and private transportation respectively. The number of commuters on different modes of transportation converged by use of learning tools and rewards. In the first day, most of the people had a tendency to use public transportation, which was decreased in the last day of the simulation period.

Total time of daily trips for both mode of transportation which was selected by agents, measured and the differences between these two times for all day long was calculated. In Fig. 3, the result is shown for the simulation period. This fluctuation was related to different factors such as traffic on road, departure time and waiting time for public transportation each day. However, in final days, the difference time between public and private transportation was less than 10 minutes by reaching equilibrium flow on transportation mode and, it seemed to be stable.

Based on rewards and decision making of departure time and transportation mode, the commuters' utilities were changed daily. Fig. 4 represents daily changes in effective utilities earned by each commuter among the whole period. In this chart, it is shown that both public and private utilities were increased with day-to-day variation.

The number of commuters on different modes, effective utility, average time of public and private and average of total time which were observed at last day are described in TABLE II , and it shows the average of total travel time of each mode were roughly similar to the average of total travel time of both modes and effective utilities of commuters had a bit difference with ones they expected.

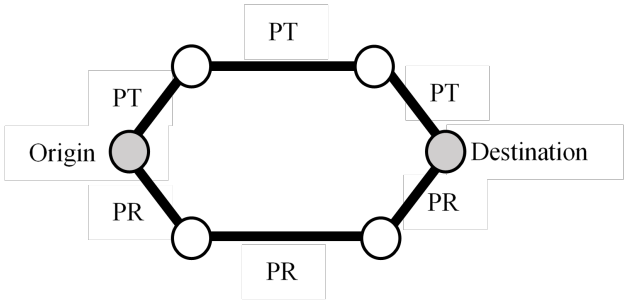


Figure 1. Network

V. DISCUSSION AND CONCLUSION

In this paper, we have proposed the framework for evaluating the reinforcement learning and effect of minority games on equilibrium flow on roads. We suggested applying agent-based modeling and simulation as a platform to implement our framework.

To illustrate, a simple network consists of two different types of mode (PT and PR) were considered and a population

Table II
THE RESULT OF THE LAST DAY

Variable	Value
No. of commuters on public transportation	102
No. of commuters on public transportation	99
Average total time on public transportation	34.26 min
Average total time on private transportation	25.14 min
Average total time of both mode	29.771 min
Average effective utility on public transportation	15.502
Average effective utility on private transportation	14.920
Average expected utility on public transportation	15.710
Average expected utility on private transportation	15.012

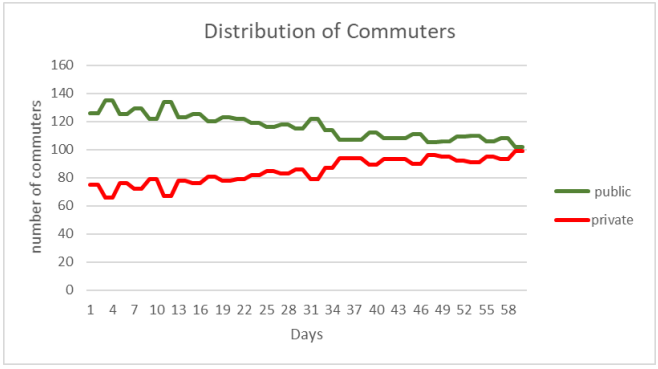


Figure 2. Distribution of commuters on roads

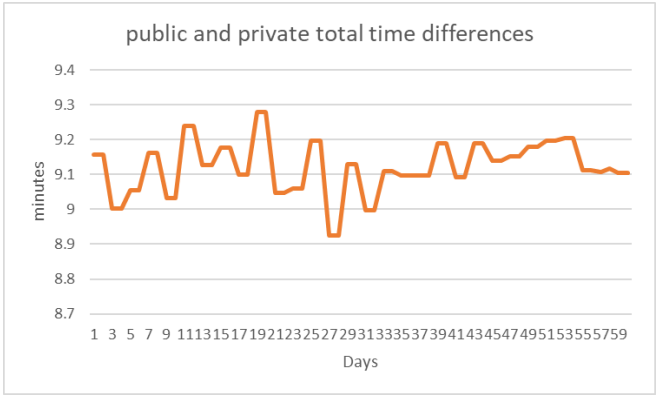


Figure 3. Public and private total time differences

of commuters with the memory of travel experiences was generated. They performed their daily plan in morning high-demand hours and their activities iterated for sixty days. Their experience, expected and effective utilities, expected and effective travel time and rewards were observed and analyzed.

In regard to results, the commuters learned to predict total travel time in both modes which their exception was similar to obtained total travel time in each mode. By balancing number of commuters on each type of transportation, they gained higher utilities rather than first days. From the illustrative example, the hypothesis of the study, which was to use RL and minority game to reach equilibrium flow, was reached and it is concluded that equilibrium flow can follow higher utilities and more precise time prediction in daily trips.

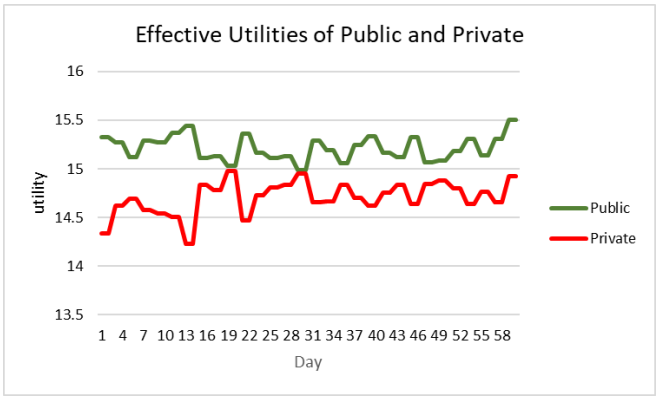


Figure 4. Effective Utilities of Public and Private

For future work, we will consider a realistic large-scale network and demands, different types of incentives and roads with combination type of transportations so as to better study and analysis of commuters behavior and performance of the transportation system. With such improvements, we are confident that our framework can be proper and accurate to increase the commuters pleasant and also the performance of the road transportation system.

REFERENCES

[1] M. Ebrahimi, License Plate Location Based on Multi Agent Systems, Intell. Eng. Syst., 2007.
[2] K. Modelewski and M. Siergiejczyk, Application of multi-agent systems in transportation, Appl. Multiagent Syst. Transp., 2013.
[3] I. Klein, N. Levy, and E. Ben-Elia, An agent-based model of the emergence of cooperation and a fair and stable system optimum using ATIS on a simple road network, Transp. Res. Part C Emerg. Technol., vol. 86, no. November 2017, pp. 183201, 2018.
[4] M. Tlig and N. Bhouri, A multi-agent system for urban traffic and buses regularity control, Procedia - Soc. Behav. Sci., vol. 20, pp. 896905, 2011.
[5] D. Grether, B. Kickhfer, and K. Nagel, Policy Evaluation in Multiagent Transport Simulations, Transp. Res. Rec. J. Transp. Res. Board, vol. 2175, no. 1, pp. 1018, 2010.
[6] J. Rouwendal and E. T. Verhoef, Basic economic principles of road pricing: from theory to applications, Transp. Policy, vol. 13, pp. 106114, 2006.
[7] E. Ben-Elia and D. Ettema, Rewarding rush-hour avoidance: a study of commuters travel behavior., Transp. Res. Part A Policy Pract., vol. 45, p. 567582., 2011.
[8] M. C. J. Bliemer and D. H. van Amelsfort, Rewarding instead of charging road users: a model case study investigating effects on traffic conditions, Eur. Transp., vol. 44, pp. 2340, 2010.
[9] A. M. Kaplan and M. Haenlein, Users of the world, unite! The challenges and opportunities of Social Media, Bus. Horiz., vol. 53, no. 1, pp. 5968, 2010.
[10] M. N. K. Grether, D; Chen, Y.; Rieser, Effects of a simple mode choice model in a large-scale agent-based transport simulation., in Complexity and Spatial Networks. In Search of Simplicity, Advances in Spatial Science, P. Reggiani, A.; Nijkamp, Ed. Springer, 2009, pp. 167186.
[11] K. Golubev, A. Zagarskikh, and A. Karsakov, A framework for a multi-agent traffic simulation using combined behavioural models, Procedia Comput. Sci., vol. 136, pp. 443452, 2018.
[12] Z. Kokkinogenis, N. Monteiro, R. J. F. Rossetti, A. L. C. Bazzan, and P. Campos, Policy and incentive designs evaluation: A social-oriented framework for Artificial Transportation Systems, 2014 17th IEEE Int. Conf. Intell. Transp. Syst. ITSC 2014, pp. 151156, 2014.
[13] J. de D. Ortzar and L. G. Willumsen, Modelling Transport. Chichester: John Wiley & Sons, Ltd, 2011.

[14] D. Challet and Y. C. Zhang, Emergence of cooperation and organization in an evolutionary game, Phys. A Stat. Mech. its Appl., vol. 246, no. 34, pp. 407418, 1997.
[15] A. Physics, The Minority Game: evolution of strategy scores, 2015.
[16] P. C. Bouman, L. Kroon, P. Vervest, and G. Marti, Capacity, information and minority games in public transport, Transp. Res. Part C Emerg. Technol., vol. 70, pp. 157170, Sep. 2016.
[17] T. Chmura, T. Pitz, and M. Schreckenberg, Minority Game - Experiments and Simulations, in Traffic and Granular Flow 03, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 305315.
[18] T. Takama and J. Preston, Forecasting the effects of road user charge by stochastic agent-based modelling, Transp. Res. Part A Policy Pract., vol. 42, no. 4, pp. 738749, May 2008.
[19] V. Nallur, E. O. Toole, N. Cardozo, and S. Clarke, Algorithm Diversity - A Mechanism for Distributive Justice in a Socio-Technical MAS, in Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, 2016, pp. 420428.
[20] Roth AE and Erev I, Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term, Games Econ. Behav., vol. 8, no. 1, pp. 164212, 1995.
[21] Uri Wilensky, Netlogo: Center for connected learning and computer-based modeling. Northwestern University, 1999.

Optimal Combination Forecasts on Retail Multi-Dimensional Sales Data

Luis Roque
Faculty of Engineering, University of Porto
HUUB AI Lab
Porto, Portugal
laroque@thehuub.co

Cristina A. C. Fernandes
HUUB AI Lab
Porto, Portugal
cafernandes@thehuub.co

Tony Silva
HUUB AI Lab
Porto, Portugal
tbsilva@thehuub.co

***Abstract*— Time series data in the retail world are particularly rich in terms of dimensionality, and these dimensions can be aggregated in groups or hierarchies. Valuable information is nested in these complex structures, which helps to predict the aggregated time series data. From a portfolio of brands under HUUB's monitoring, we selected two to explore their sales behaviour, leveraging the grouping properties of their product structure. Using statistical models, namely SARIMA, to forecast each level of the hierarchy, an optimal combination approach was used to generate more consistent forecasts in the higher levels. Our results show that the proposed methods can indeed capture nested information in the more granular series, helping to improve the forecast accuracy of the aggregated series. The Weighted Least Squares (WLS) method surpasses all other methods proposed in the study, including the Minimum Trace (MinT) reconciliation.**

I. INTRODUCTION

There are numerous potential benefits of using forecasting models on fashion companies, such as a reduction of the bullwhip effect, a possible reduction of the difficulties of the supplier production, an effectiveness improvement of the sourcing strategy of the retailer, the reduction of lost sales and markdowns and, consequently, the increase of profit margins, as described by [13].

Fashion markets are, however, volatile and hard to predict. [13] points out the different perspectives and requirements to bear in mind when developing a forecasting model to suit its specificities. There are two main forecast horizons to consider: medium term to plan sourcing and production and short term to plan replenishment. The products life cycle is very short in the fashion industry and there are some additional constraints such as "one shot" supply [13]. This shows the relevance of forecasting in the purchase quantities determination. At the same time, there are basic and best selling items with a completely different life and purchasing cycles. Due to these short life cycles, historical data of the disaggregated product structure are ephemeral, hence the importance of forecasting different levels of aggregation. The strong seasonality also arises as one of the important behaviours to model. The sales data generated by companies is therefore highly dimensional, meaning that it is possible to aggregate and disaggregate it by a significant number of dimensions, e.g product structure, geography, etc.

Several studies show that forecasting the aggregates by disaggregates results in better forecasts than using the best individual models [5], [3], [17]. The most common approach to leverage the information present in each aggregation level, hierarchy or group, was initially to use the top-down or the bottom-up approaches. [8] proposed a new method for combining different levels coherently across the aggregation structure, resulting in what they designated *optimal forecasts*, which proved better than the standard methods. The work by [3] was one of the first applications of the hierarchical forecasting model and its focus was on tourism demand time series data. The data were disaggregated by geographical region and by purpose of travel, thus forming a natural hierarchy of time series to reconcile. The potential of this optimal combination method was demonstrated immediately in the first attempt, surpassing the bottom-up and all the top-down approaches, with the exception of the top-down approach based on forecast proportions. Later, [11] presented improvements in the method and proved that the optimal combination method actually outperformed all the traditional methods.

Another important discussion in the field is the comparison between statistical models and machine learning. [16] shared the findings on the M3 competition, where a large subset of 1045 monthly time series was used, and statistical methods dominated across all the accuracy measures and forecasting horizons. More recently, the M4 competition has shown the potential of hybrid and combination models, using both statistical and machine learning models, which dominated the top 10 of the competition [15].

Our work is focused on statistical models to build solid baseline forecasting models and optimally reconcile them using a hierarchical approach, but also sustained by the work being done in the field, where machine learning models still do not match the performance of statistical models when only a pure time series is provided.

Section II describes the theory and previous studies related to the methodologies implemented in this work. Section III lists the performance measures used to analyse and compare our models. In Section IV we present the dataset in study and in Section V we describe the methodology applied. Our results and discussion are detailed in Section VI and our conclusions and future work are summarised in Section VII.

II. RELATED WORK

A. Data Pre-processing

Different approaches can be taken to pre-process time series data in order to stationarize, normalize, detrend and

deseasonalize data. Pre-processing is an important stage as it has been shown that forecasting models perform better on pre-processed data [15]. Indeed, a time series will be easier to model accurately the more stationary and closer to a normal distribution it is. For a time series to be stationary, its properties, such as mean, variance and autocorrelation, should be constant over time. Various transformations can be applied to data for this purpose. The Box-Cox transformation is used to modify the distributional shape of a set of data to stabilize variance, make the data more normal distribution-like so that tests and confidence limits that require normality can be appropriately used. It is defined by:

$$y^{(\lambda)} = \begin{cases} (y^{\lambda}-1)/\lambda & ,\text{if } \lambda \neq 0 \\ \log(y) & ,\text{if } \lambda = 0 \end{cases} \quad (1)$$

where λ is the transformation power parameter and y is a list of strictly positive numbers. Hence, the Box-Cox transformation can only be applied to positive data.

A Box-Cox transformation with $\lambda = 0$ is equivalent to a simple logarithmic transformation.

B. Model description

The Autoregressive Integrated Moving Average (ARIMA) or Box Jenkins methodology is one of the most popular and frequently used stochastic time series models. The time series is assumed to be linear and follows a particular known statistical distribution, such as the normal distribution, which can make it inadequate in some practical situations [1]. However, its flexibility to represent several varieties of time series with simplicity and the optimization of the model building process render the Box-Jenkins methodology as a well-suited baseline approach [1].

In an AR(p) model, the future value of a variable is assumed to be a linear combination of p past observations and a random error together with a constant term [1]. Mathematically:

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t, \quad (2)$$

where y_t is the model value; ε_t is the random error (or random shock) at time period t; ϕ_i are model coefficients and c is a constant; p is the order of the model, also called the *lag*.

The MA(q) model uses past errors as the explanatory variables [1]. Mathematically:

$$y_t = \mu + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t, \quad (3)$$

where μ is the mean of the series; θ_j are the model parameters; q is the order of the model. The random errors, or random shocks, are assumed to be white noise, i.e. independent and identically distributed (i.i.d.) random variables with zero mean and a constant variance σ^2 [1].

ARMA is a combination of the autoregressive and moving average models. Matematically:

$$y_t = c + \varepsilon_t + \sum_{i=i}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \quad (4)$$

The ARMA model can only be used for stationary time series data. However, in practice many time series such as those related to socio-economy and business show non-stationary behaviour. Time series, which contain trend and seasonal patterns, are also non-stationary in nature. For this reason, the ARIMA model is proposed.

The ARIMA model is a generalization of an ARMA model to include the case of non-stationarity. The non-stationary time series are made stationary by applying finite differencing of the data points. Mathematically it can be written as:

$$\begin{aligned} &(1-\sum_{i=1}^{p} \phi_i L^i)(1-L)^d y_t = (1+\sum_{j=1}^{q} \theta_j L^j)\varepsilon_t \\ &\Leftrightarrow \phi(L)(1-L)^d y_t = \theta(L)\varepsilon_t \end{aligned} \quad (5)$$

where L is the lag or backshift operator, defined as $Ly_t = y_{t-1}$ and

$$\begin{aligned} \phi(L) &= 1 - \sum_{i=1}^{p} \phi_i L^i \\ \theta(L) &= 1 + \sum_{j=1}^{q} \theta_j L_j, \end{aligned} \quad (6)$$

with p, d and q being integers that refer to the order of the autoregressive, integrated and moving average parts of the model, respectively. The integer d controls the level of differencing.

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model is the generalization of the ARIMA model for seasonality. In this model, the seasonality is removed by seasonally differencing the series. In other words, by computing the difference between one observation and the corresponding observation from the previous year (or season): $z = y_t - y_{t-s}$. For a monthly time series, $s = 12$, for a quarterly time series $s = 4$, etc.

Mathematically, a SARIMA $(p,d,q) \times (P,D,Q)^s$ model is formulated as:

$$\Phi_P(L^s)\ \phi_p(L)\ (1-L)^d\ (1-L)^D\ y_t = \Theta_Q(L^s)\ \theta_q(L)\ \varepsilon_t \quad (7)$$

where P, D, Q are the seasonal orders of the autoregressive, integrated and moving average parts of the model, respectively, and s is the seasonal period.

C. Model Evaluation

Good models are obtained by minimising the Akaikes Information Criterion (AIC), the Corrected Akaikes Information Criterion (AIC_c) or the Schwarzs Bayesian Information Criterion (BIC). The BIC and the AIC_c were developed as a bias-corrected version of the AIC, better fit for short time series, where the AIC tends to select more complex models with too many predictors. [9] indicates its preference to use the AIC_c to compare models.

A note should be made about the fact that the AIC, AIC_c and BIC criteria should not be used to compare the performance of models with different differencing orders. The reason for this is that these criteria are all based on the likelihood of the model. This likelihood will be different for a series that is differenced and for a series that is not differenced. These criteria can only directly compare models with the same seasonal and first differencing orders.

If there are significant correlations between different lags of the data, or the residuals of a model, this indicates that there are still important features in the data that the model is not reproducing well and thus there is valid information to learn. [14] proposed the Ljung-Box (LB) test to evaluate the overall randomness of data based on a number of lags. It is commonly used to check if the residuals from a time series resemble white noise, i.e. if the relevant information was captured by the model. It is based on the statistic Q^*, which can be written as:

$$Q^* = T(T+2)\sum_{k}^{l}(T-k)^{-1}r_k^2, \tag{8}$$

where T is the length of the time series, r_k is the k^{th} autocorrelation coefficient of the residuals and l is the number of lags to test. Q^* approximates a chi-squared distribution with $l-K$ degrees of freedom, where K is the number of parameters of the estimated model. Large values of Q^* indicate that there are significant autocorrelations in the residual series.

The null hypothesis of the LB test states that the data are independently distributed, not showing serial correlation. A significant p-value in this test thus rejects the null hypothesis. Conversely, small values of p-value indicate the possibility of non-zero autocorrelations.

D. Hierarchical and Grouped time series

Univariate Time Series data rely only on time to express patterns. When there is aggregated data available, there are potential covariance patterns nested in the hierarchy. [6] summarizes guidelines on using the more traditional hierarchical forecasting approaches, top-down, bottom-up and a combination of both, the middle-out approach. Among these three, top-down and middle-out rely on a unique hierarchy to assign the weights from the higher aggregated series to the lower ones, so it is not suitable for grouped time series. Grouped time series are built based on a structure that disaggregates based on factors that can be both nested and crossed [9].

A group can have several levels, starting on the most aggregated level of the data and disaggregating it by attributes, e.g. A and B, forming the series $y_{\mathrm{A},t}$ and $y_{\mathrm{B},t}$, or by a second structure of attributes, e.g. X and Y, forming series $y_{\mathrm{X},t}$ and $y_{\mathrm{Y},t}$. At the bottom level, this would generate eight different series, $y_{\mathrm{AX},t}$, $y_{\mathrm{AY},t}$, $y_{\mathrm{BX},t}$, $y_{\mathrm{BY},t}$, $y_{\mathrm{XA},t}$, $y_{\mathrm{XB},t}$, $y_{\mathrm{YA},t}$ and $y_{\mathrm{YB},t}$.

[8] proposed a new method to output more coherent forecasts, adding them up consistently within the aggregation structure. This method also proposes a generalized representation, from which the earlier approaches are obtained as special cases. As it is based on the reconciliation of the independent forecasts of all the aggregation levels, the reconciled forecasts are always more consistent than when a bottom-up strategy is used. This method is explored in more detail in the following section.

E. Optimal Forecast Reconciliation

The method consists in optimally combining and reconciling all forecasts at all levels of the hierarchy. To combine the independent forecasts, a linear regression is used to guarantee that the revised forecasts are as close as possible to the independent forecasts while maintaining coherency. The independent forecasts are reconciled based on:

$$\hat{y}_T(h) = S\beta_T(h) + \varepsilon_h, \tag{9}$$

where $\hat{y}_T(h)$ is a matrix of h-step-ahead independent forecasts for all series, stacked in the same order as the original data, S is a summing matrix, $\beta_T = \mathrm{E}[b_{T+h}\,|\,y_1,...,y_T]$ is the unknown mean of the most disaggregated series at the bottom level, and ε_h represents the reconciliation errors. [8] also proved that the resulting forecasts are unbiased, which is not the case in the top-down approach.

[11] proposed a weighted least-squares (WLS) reconciliation approach, concluding that it outperforms the ordinary least squares (OLS). In their study, the only exception was the very top level where the OLS method did slightly better. The notation for the reconciliation approach was presented differently in [12], referring to the new method as the MinT (minimum trace) reconciliation, represented as:

$$\tilde{y}_T(h) = S(S^T W_h^{-1} S)^{-1} S^T W_h^{-1} \hat{y}_T(h), \tag{10}$$

where $\tilde{y}_T(h)$ is the reconciled h-step ahead forecast and W_h is the variance-covariance matrix of the h-step-ahead base forecast errors. As it is challenging to estimate W_h, [12] proposed five different approximations for estimating this parameter, some of which we implemented in our work.

Besides the MinT approach, we considered the approximation case where it is assumed that $W_h = k_h I, \forall h$, where $k_h > 0$ and I is the identity matrix. In reality, this assumption collapses MinT to the early proposal of [8], to use a simple OLS estimator. This is a very strong assumption, where the base forecast errors are considered uncorrelated and equivariant, which is not possible to satisfy in hierarchical and grouped time series.

We also considered the case where MinT can be described as a WLS estimator [12], which can be written as $W_h = k_h \mathrm{diag}(\hat{W}_1)$, for all h, where $k_h > 0$, and

$$\hat{W}_1 = \frac{1}{T}\sum_{t=1}^{T} e_t e_t', \tag{11}$$

where e_t is a vector of the residuals of the models that generated the base forecasts stacked in the same order as the data. This method scales the base forecasts using the variance of the residuals, hence it works as a weighted least squares estimator using variance scaling.

III. PERFORMANCE MEASURES

A. Mean Absolute Scaled Error (MASE)

MASE is a measurement of forecast accuracy which scales the errors based on the *naïve* forecast method. If MASE is smaller than 1, the forecast method is better than the average one-step naive forecast. MASE is scale-free so can be used to compare different forecast accuracies. Its mathematical expression is given by:

$$MASE = \text{mean}(|q|) = \frac{1}{T}\sum_{t=1}^{T}|q|, \tag{12}$$

where q is defined as:

$$q = \begin{cases} \dfrac{|\hat{y}_t - y_t|}{\dfrac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}|}, \text{ if } \{y_t\} \text{ is non-seasonal} \\ \dfrac{|\hat{y}_t - y_t|}{\dfrac{1}{T-s}\sum_{t=s+1}^{T}|y_t - y_{t-s}|}, \text{ if } \{y_t\} \text{ is seasonal,} \end{cases} \tag{13}$$

where s is the seasonal period and T is the length of the time series.

B. Root Mean Squared Error (RMSE)

RSME is one of the most used metrics to evaluate forecast models. As it is not normalized, it can only be used to compare models in the same dataset. It is given by:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}e_t^2}. \tag{14}$$

IV. DATASET

HUUB is a logistics company for the fashion industry. It offers a platform for fashion brands that integrates the full supply chain process in one place, simplifying operations from production to customer service. HUUB's platform makes use of data to boost business growth and forecasting is a key aspect of its analysis. It has today over 50 brands under its monitoring, selling in 4 different channels: eCommerce, marketplaces, stores and retail.

The time series of all HUUB brands have been analyzed in order to explore their sales behaviour. Only brands with more than two years of data and consistent sales on a weekly basis were considered for forecasting analysis. For an initial study, we selected only the eCommerce sales channel of two of the largest and most representative brands, ranked in the top 10. The aim of this data reduction is to narrow the scope of the problem, since the purpose is to have a first approach and a strong baseline for the future. Furthermore, combining only forecasts of the two most prominent brands helps to improve the forecast performance.

Figure 1 plots the selected eCommerce brands data series. The data is aggregated in weeks and ranges from the 11-12-2016 (Sunday), which is the earliest week for which we have complete data of all series included in the analysis, to the 19-01-2019 (Saturday).

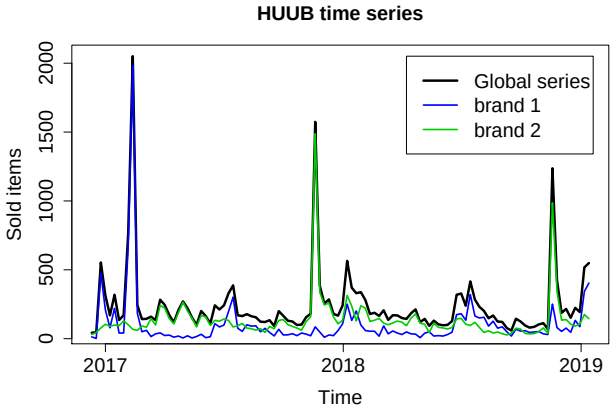


Fig. 1: Time series profile of HUUB eCommerce selected brands.

V. METHODS

A. Models specification

[4] proposed a practical and pragmatic approach to build ARIMA models. [2] defined, based on that methodology, 4 iterative steps, for the univariate time series SARIMA model.

1) Model identification: This step involves the selection of parameters to include in the model, both seasonal and non-seasonal, based on the Autocorrelations function (ACF) and the Partial Autocorrelations function (PACF).
2) Parameter estimation: the previously identified parameters are then estimated using Least Squares or Maximum Likelihood and a first selection of models is conducted using information criteria.
3) Diagnosis of the fitness of the model: The model is diagnosed using the statistical properties of the error terms, using Ljung-Box statistic test to check the adequacy. When the errors are normally distributed, we can move towards the next step.
4) Forecasting and validation: The model is validated using out-of-sample data and applied to forecast the future values.

B. Grouped Time Series

The structure used for this grouped time series starts at the top level with the time series that sums up all series at lower levels with different attributes. It is disaggregated by brand, *brand 1* and *brand 2*, in a hierarchical structure. Besides this, we have five different representations of the aggregation, since there are five attributes to consider in this study: gender (Male, Female or Unisex), age group (Baby, Kid, Baby Kid, Teen or Adult), product type (Clothing, Footwear, Accessories, Homewear, Beachwear, Swimwear, Swim accessories, Compound, Stationery, Underwear, Nightwear or Sports), season of the product (Seasonal or Permanent). Figure 2 shows one of the alternatives to represent the group structure. It generates a total of 128 time series to be reconciled. When grouping time series, the complexity increases quickly, which requires significantly more computation resources to solve the problem. For each of the time series, we applied the best possible model to forecast that individual series using

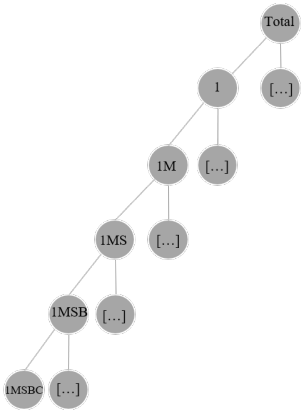


Fig. 2: A representation of one of the five level grouped structures: *Total*, the global time series; *brand 1*, (1), and *brand 2*, (2); *brand 1* crossed with gender *male* (1M); *brand 1* crossed with gender *male* and *seasonal* items (1MS);*brand 1* crossed with gender *male*, *seasonal* items and age group *baby* (IMSB); and, finally, *brand 1* crossed with gender *male*, *seasonal* items, age group *baby* and product type *clothing* (IMSBC).

the *auto.arima()* function from the R *Forecast* package [10], [7].

VI. RESULTS

A. Baseline model - SARIMA

Based on the mathematical expression (7) for the SARIMA function, we can deduce some parameters restrictions according to the size of the time series window. For instance, if we have a time series of length 114 data points, and we choose a SARIMA model with parameters $(0,0,0)(2,1,0)^{52}$, this is an unfeasible model since it would be a linear combination of past data points given by:

$$y_t = (1+\Phi_1)\,y_{t-52} + (\Phi_2-\Phi_1)\,y_{t-104} - \Phi_2\,y_{t-156} + \varepsilon_t, \quad (15)$$

and we do not have data for time period $t-156$, as would be required to compute the model.

By working out the mathematical expression of the SARIMA model (Eq. 7) we can find the restrictions to the orders of the SARIMA model parameters as a function of the insample window size which is used to produce the forecasts. A forecast produced by a SARIMA model of parameters $(p,d,q)(P,D,Q)^s$ will only be feasible if the following conditions are met:

$$\begin{aligned} &(D+P)\,s+p+d \leq \text{window size} \\ &Q\,s+q \leq \text{window size}. \end{aligned} \quad (16)$$

Moreover, the best coefficients for each SARIMA model are found by fitting the model to the time series and in order to have all the data points to compute them, the time series should have at least one full seasonal period for which the past data points required to compute the series are all available. As an example, if we want to attempt fitting a SARIMA$(1,1,0)(0,1,0)^{52}$ model to a time series of length T, we will need at least $(D+P)s+p+d=54$ data points on top of a full seasonal period in order to have all the data points required to compute the model in at least one full seasonal period. If this is not the case, there will be some missing data and the model coefficients will not be as well-fit to the data as possible. The criteria used to estimate the quality of the model, such as the AIC and BIC, will not be well estimated in that case as well, given the missing data.

Therefore, the more general restrictions for the SARIMA model as a function of the time series length are:

$$\boxed{\begin{aligned} &(D+P+1)\,s+p+d \leq T \\ &(Q+1)\,s+q \leq T \end{aligned}} \quad (17)$$

Our time series has a length of $T=110$ data points (weeks) and a seasonal period of 52 weeks. We want to test a forecasting of 4 weeks, so we divide the time series into 106 data points for finding the best SARIMA parameters and coefficients, the training series, and 4 data points to test the forecast retrieved by that model, the testing series. The length of the training series limits our SARIMA parameters in the following way: $(D+P)\,s+p+d \leq 54$ and $Q\,s+q \leq 54$.

Brand 1

The sales of *brand 1* vary a lot with time, which reflects in a time series with a high variance (see Figure 1). This suggests that a Box-Cox transformation can make it more stationary.

The optimal λ parameter can, in principle, be determined by ensuring that the standard deviation of the transformed data is minimum, which is what the *BoxCox()* function from the R *Forecast* package ([10], [7]) with *lambda="auto"* computes. Even though there is a high likelihood that the data will be normally distributed after a Box-Cox transformation with *lambda="auto"*, this is not guaranteed. A quantile-quantile (Q-Q) plot shows the distribution of the data against the expected normal distribution, hence it is an effective diagnosis to determine whether a sample is drawn from a normal distribution. Figure 3 shows the Q-Q plot of the Box-Cox transformed data with *lambda="auto"* as well as the Q-Q plot of a simple logarithmic function, $\lambda=0$. The more data points fall in a straight line, the more normally distributed they are. If they deviate from a straight line in any systematic way, this suggests that the data is not drawn from a normal distribution. Figure 3 thus indicates that the logarithmic transformation effectively approximates the data to a normal distribution more than the Box-Cox transformation with *lambda="auto"*. A $\lambda=0$ Box-Cox transformation was therefore applied to the data.

Figure 4 presents *brand 1* 1 time series after Box Cox transformation. This series displays a clear seasonal pattern with a cyclic profile with a yearly, or half-yearly, period. A seasonal differencing with a lag of 52 weeks (1 year) or 26 weeks (half-year) should, thus, improve the time series. Indeed, the time series standard deviation decreases from 1.053 to 1.020 and to 1.028, after applying a 52-weeks and a 26-weeks differencing, respectively, which improves the series in both cases. A first differencing of the series increases the standard deviation and a unit root test also returns no first differencing, it is thus not expected to improve the modelling of the series.

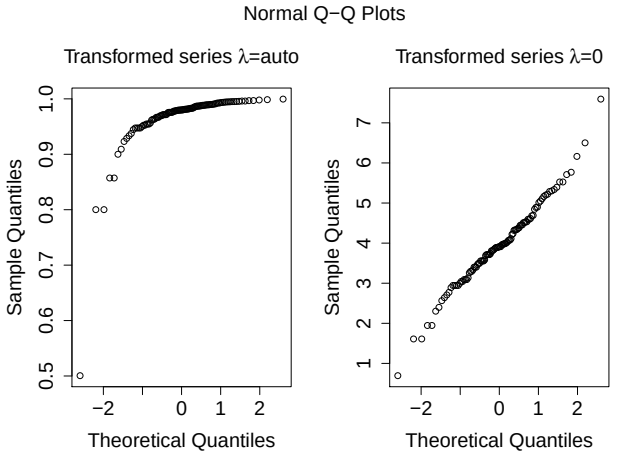


Fig. 3: (Left:) Q-Q plot of *brand 1* 1 Box-Cox transformed time series with $\lambda =$ "*auto*"; (Right:) Q-Q plot of *brand 1* 1 Box-Cox transformed time series with $\lambda = 0$, equivalent to a logarithmic transformation. It is clear that the latter series is closer to a normal distribution, hence it is a better transformation.

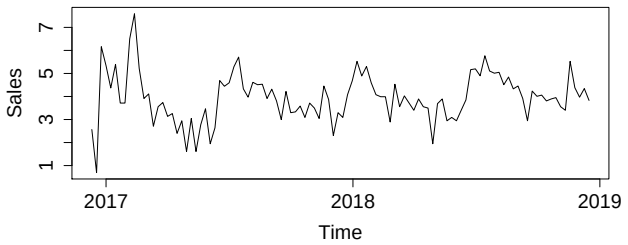


Fig. 4: *Brand 1* time series after Box Cox transformation with $\lambda = 0$.

The analysis of the ACF and PACF of the yearly differenced series indicate that a SARIMA$(0,0,1)(0,1,0)^{52}$ model is a good candidate to reduce the data autocorrelations. This and other SARIMA models were tested and this was the model with the lowest AIC_C and BIC criteria. A more complex model, SARIMA$(6,0,0)(0,1,0)^{52}$, was favoured by the AIC criteria, however, according to the restrictions in (17), this model exceeds the threshold limited by the size of the training series. Moreover, particularly when the sample size is small, the AIC criteria favours higher order (more complex) models as a result of overfitting the data. The AIC_C is a correction of the AIC and a more realistic criteria for smaller sample sizes, along with the BIC criteria. For these reasons, and since simpler models tend to be a better choice for accurately forecasting the general behaviour of time series, we chose $(0,0,1)(0,1,0)^{52}$.

The half-year seasonally differenced series was also analysed and we found SARIMA$(0,0,0)(0,1,1)^{26}$ to be the best fit model. Its RMSE was lower than most SARIMA models with a 52-weeks period, which could indicate that its forecast was better, but the AIC, AIC_C and BIC were significantly higher, implying that it provides a worst fit to the time series. The Ljung-Box test of the residuals of these models yielded lower p-values than for 52-weeks period models, signaling that they are worst models. For these reasons, we consider that the best SARIMA baseline model for *brand 1* is SARIMA$(0,0,1)(0,1,0)^{52}$.

Brand 1 had a major peak of sales in the first quarter of 2017 which did not repeat in 2018 and which was almost 10 times above the average of the subsequent peaks (see Figure 1). This sales peak was due to strong campaigning for an online collection launching. It was a one-time event and the peak affects the SARIMA models, increasing its prediction for the same period of subsequent years. We performed the same analysis having this outlier peak removed and its values interpolated. The time series forecasting RMSE was reduced by 22%. However, to maintain the method's automation capacity, we chose not to implement any outlier removal. Instead, external variables will be used to handle this type of event in a subsequent article.

Brand 2

The time series for *brand 2* has a seasonal profile, which is also clear in the ACF and PACF at lag 52 (see Figure 5). Since the series shows variations which change with time, a transformation can improve the series. Similarly to *brand 1*, the Box-Cox transformation with $\lambda = 0$ improved the normalization of the time series over *lambda* = "*auto*".

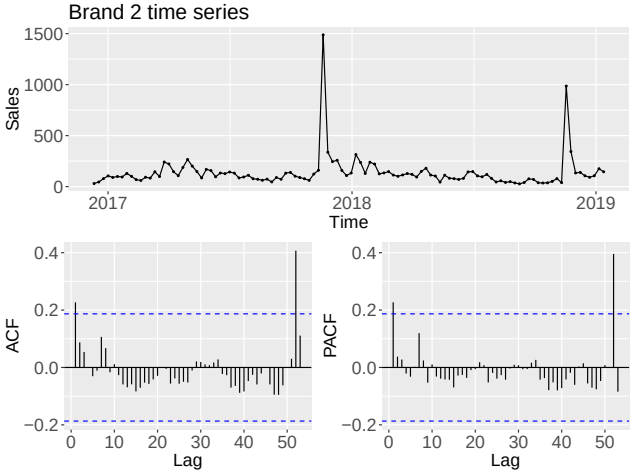


Fig. 5: *Brand 2* original time series and respective ACF and PACF

Since *brand 2* displays a clear seasonal profile, it is not stationary and a seasonal difference can improve the series. Even though the standard deviation of the series increases from 0.64 to 0.68 after seasonal differencing, the series becomes more normalized.

The series is also more stationary after a first difference and the standard deviation decreases to 0.54. A second difference would increase significantly the standard deviation, so no more differencing was applied to the series.

The ACF and PACF of the differenced series show a significant negative lag 1 of ≈ -0.3, which suggests a mild overdifferencing. This can be counterbalanced with one order of a moving average model applied to the series, which indicates that SARIMA$(0,1,1)(0,1,0)^{52}$ is a good candidate model for the data.

This and other models, including different differencing orders, were tested and the BIC criteria, which favours

simpler model, was used to compare them. The BIC criteria also favoured SARIMA$(0,1,1)(0,1,0)^{52}$ model out of all models.

The residuals of the SARIMA$(0,1,1)(0,1,0)^{52}$ model applied to *brand 2* time series show all autocorrelation lags below the threshold, apart from lag 16, which corresponds to roughly a trimester and it is thus, most likely, related to the periodic behaviour of the brand. The fact that the residuals still display significant autocorrelations is also translated in a relatively low value of the LB p-value$= 0.12$. As more data becomes available and are included in the time series, it will become more obvious whether the data autocorrelation at lag 16 is real, and, if so, it will be easier to model it.

Global time series

A global time series is formed by the sum of *brand 1* and *brand 2* time series. The same pre-processing and analysis performed on *brand 1* and *2* time series were applied to the global time series, in the same 4 weeks period range.

As expected, the global time series also exhibits a seasonal profile, with a cyclic behaviour and significant autocorrelations at lags 52. This entails a seasonal differencing, hence $D = 1$. The standard deviation of the time series increases when applying a first difference and the unit root test determined that the time series is stationary so needs no first differencing, both advocating for $d = 0$. The model that presented lower AIC, AIC$_c$ and BIC criteria was SARIMA$(0,0,1)(0,1,0)^{52}$.

The residuals of SARIMA$(0,0,1)(0,1,0)^{52}$ showed no significant autocorrelations at any lag, which indicates that SARIMA$(0,0,1)(0,1,0)^{52}$ is a well-fit model for the global time series. Its residuals and the high LB p-value of 0.67 indicate a reasonable resemblance of residuals with white noise, further approving this model.

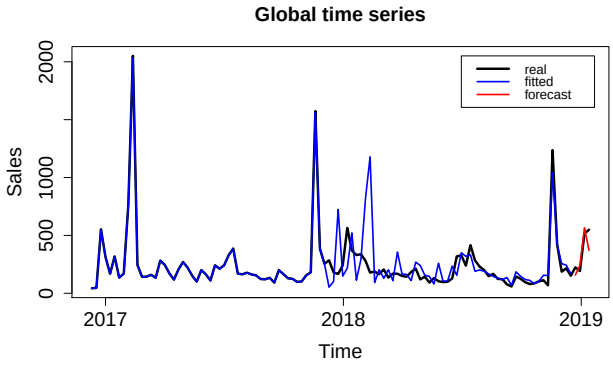


Fig. 6: Global time series (black) with SARIMA$(0,1,1)(0,1,0)^{52}$ model fitting (blue) and 4-weeks forecasting (red).

The fitting and forecast of the global time series by SARIMA$(0,0,1)(0,1,0)^{52}$ are shown in Figure 6.

Baseline models performance

The baseline models performance summary is presented in Table I

TABLE I: Baseline SARIMA models for *brand 1*, *brand 2* and the *global* time series. The time period of forecast was 4 weeks.

Data	SARIMA	AIC	AIC$_c$	BIC	LB p-value	MASE	RMSE
Global	$(0,0,1)(0,1,0)^{52}$	101.05	101.28	105.02	0.67	0.69	101.16
Brand 1	$(0,0,1)(0,1,0)^{52}$	153.72	153.95	157.70	0.80	1.17	145.79
Brand 2	$(0,1,1)(0,1,0)^{52}$	77.23	77.47	81.17	0.12	0.24	19.48

B. Grouped Time Series Forecast Reconciliation

As explained in Section V, the five different product attributes selected generated 128 different time series to reconciliate. The best possible model parameters and coefficients were determined for each of these individual series, using the *auto.rima()* function [10], [7]. The best individual forecasts were used as input for the reconciliation process. Table II shows the performance metrics in the out-of-sample data: independent (base) forecasts, used as baseline, bottom-up and the three approaches considered as the optimal reconciliation methods (OLS, WLS and MinT, proposed by [11] and [12]).

All approaches were tested using a horizon h of 4 weeks, which was identified by HUUB as the most relevant period range to be forecasted, given the weekly data granularity. The results are presented for the three most aggregated time series, despite the fact that the grouped time series yielded results for all the possible combinations. This is also a major advantage for the business, allowing to drill down to more granular dimensions, with the reassurance that the forecasts are consistent across these dimensions. The decision making process can be significantly empowered, for instance when defining purchase quantities for the next collection for specific product type, age group, etc.

Overall, the results show that the best performing method for each series can leverage nested information in the group structure and add consistency to the outcome. This is supported by the RMSE of WLS in *brand 1*, which has a very significant improvement when compared to the baseline. In terms of *brand 2*, WLS results were very close to the baseline, which can be explained by the simpler grouping structure for this brand, i.e. it is essentially concentrated in one product type, one gender, one age group, one season. For the *global* time series, the baseline model performed slightly better than the reconciled best model, with an RMSE 5% lower than WLS. Overall, the RMSE indicates that WLS is a robust method for the forecasting of seasonal time series when no further information is known about what causes the time series variations.

The best performing model in terms of MASE metric is the simpler OLS method for *brand 1* and *global* time series. For *brand 2*, MASE is lower for the baseline model. The MASE error of the WLS is lower than that of the baseline for the *global* and *brand 1* series, and very similar for *brand 2*.

Adding more brands and more attributes can, in principle, help to increase the accuracy of the proposed models relative to the baseline and, indeed, preliminary work with a disaggregation with more attributes is showing a significant improvement.

Examining the proposed methods, the one that yielded

better results in terms of RMSE was the WLS, outperforming the very ineffective bottom-up, but also the simpler OLS and the new MinT approach.

TABLE II: Models performance in the out-of-sample data of the baseline forecasts, bottom-up and the approaches considered as the optimal reconciliation methods: OLS, WLS and MinT. The best performing models according to the RMSE are highlighted in bold.

Data	Model	MASE	RMSE
Global	**Baseline**	**0.69**	**101.16**
	Bottom-up	1.16	180.93
	OLS	0.63	108.08
	WLS	0.67	105.96
	MinT	0.72	109.51
Brand 1	Baseline	1.17	145.79
	Bottom-up	1.74	189.41
	OLS	0.81	113.21
	WLS	**0.86**	**107.72**
	MinT	0.94	110.21
Brand 2	**Baseline**	**0.24**	**19.48**
	Bottom-up	0.31	26.07
	OLS	0.30	22.33
	WLS	**0.26**	**19.55**
	MinT	0.25	20.18

VII. CONCLUSIONS AND FUTURE WORK

This work explored the optimal forecasts method for hierarchical and grouped time series proposed by [8], [12] to forecast the weekly sales of two fashion brands under HUUB's monitoring. The data were disaggregated according to five selected attributes: brand, gender, whether the item is seasonal or permanent, age group and product type. This disaggregation resulted in a grouped structure with 128 individual time series. These series were individually fit by the best SARIMA model found by the *auto.arima()* function and were then reconciled following the optimal forecasts method. Three difference cases of the optimal forecasts method were explored: OLS, WLS and MinT, as well as the simple Bottom-up approach.

The forecasts performance of the three most aggregate series was evaluated and compared to the forecasts obtained by baseline SARIMA models. Each of these baseline SARIMA model was found by manually analysing the time series ACF, PACF and through stationarity and differencing tests in order to determine the best model parameters and coefficients.

The comparison of the performance of the optimal forecasts method and the baseline SARIMA models showed that WLS was the forecast method with the lowest RMSE for the two brands time series. The global time series was slightly better forecasted by the baseline model. Overall, the RMSE indicates that WLS is a robust method for the forescasting of seasonal time series when no further information is known about what causes the time series variations. Preliminary work with a higher-level disaggregation with more attributes has shown a significant improvement of the RMSE of the global time series forecast. These findings will be presented in a subsequent article.

Out of the three explored methods, WLS was the one with the highest RMSE accuracy followed by MinT and then OLS. All of them performed better than the less efficient bottom-up method, as expected.

The proposed work was intentionally focused on univariate time series models, providing a strong baseline and framework to build more complex forecasting models. Future work will contemplate the addition of more brands and more attributes (not only product attributes but others like sales markets geography structures), increasing the capacity of the model to capture even more nested information in these granular series. One possibility is the use of hybrid models, that combine statistics and machine learning, already proven to generate better results than statistical ones [15].

Another aspect that our work will exploit is the use of multivariate models, since they can add significant information on the volatile and unexpected behaviour observed in the data.

REFERENCES

[1] Ratnadip Adhikari and R. K. Agrawal. An introductory study on time series modeling and forecasting. *CoRR*, abs/1302.6613, 2013.

[2] Nari Sivanandam Arunraj, Diane Ahrens, and Michael Fernandes. Application of SARIMAX model to forecast daily sales in food retail industry. *IJORIS*, 7(2):1–21, 2016.

[3] George Athanasopoulos, Roman A Ahmed, and Rob J Hyndman. Hierarchical forecasts for australian domestic tourism. *International Journal of Forecasting*, 25(1):146–166, 2009.

[4] George EP Box and Gwilym M Jenkins. *Time series analysis: forecasting and control, revised ed*. Holden-Day, 1976.

[5] Carlos Capistrán, Christian Constandse, and Manuel Ramos-Francia. Multi-horizon inflation forecasts using disaggregated data. *Economic Modelling*, 27(3):666–677, 2010.

[6] Gene Fliedner. Hierarchical forecasting: issues and use guidelines. *Industrial Management & Data Systems*, 101(1):5–12, 2001.

[7] Rob Hyndman, George Athanasopoulos, Christoph Bergmeir, Gabriel Caceres, Leanne Chhay, Mitchell O'Hara-Wild, Fotios Petropoulos, Slava Razbash, Earo Wang, and Farah Yasmeen. *forecast: Forecasting functions for time series and linear models*, 2018. R package version 8.4.

[8] Rob J Hyndman, Roman A Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, 55(9):2579–2589, 2011.

[9] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[10] Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3):1–22, 2008.

[11] Rob J Hyndman, Alan J Lee, and Earo Wang. Fast computation of reconciled forecasts for hierarchical and grouped time series. *Computational Statistics & Data Analysis*, 97:16–32, 2016.

[12] Shanika L Wickramasuriya, George Athanasopoulos, and Rob Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, pages 1–45, 03 2018.

[13] Radim Lenort and Petr Besta. Hierarchical sales forecasting system for apparel companies and supply chains. *Fibres and Textiles in Eastern Europe*, 21(6):7–11, 2013.

[14] G. M. LJUNG and G. E. P. BOX. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

[15] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, 13(3):1–26, 03 2018.

[16] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018.

[17] Han Lin Shang and Steven Haberman. Grouped multivariate and functional time series forecasting: An application to annuity pricing. *Insurance: Mathematics and Economics*, 75:166–179, 2017.

SESSION 3

Telecommunications

Performance Evaluation of Routing Protocols for Flying Multi-hop Networks

André Coelho
INESC TEC and Faculdade de Engenharia, Universidade do Porto, Portugal
{ee11141}@fe.up.pt

***Abstract*—The advent of small and low-cost UAVs paved the way to the usage of swarms of UAVs that cooperate between themselves. However, cooperation requires reliable and stable communications between the UAVs, giving rise to Flying Multi-hop Networks (FMNs). FMNs introduce new routing challenges, including frequent changes in the network topology and in the quality of the radio links. FMNs are also being used to provide Internet access in remote areas where a network infrastructure does not exist, and to enhance the capacity of existing networks in temporary crowded events. These scenarios bring up even more significant routing challenges, in order to attend the users' traffic demands. Nevertheless, the application of existing ad-hoc routing protocols in FMNs remains not well characterized.**

This paper presents a performance evaluation of state of the art routing protocols applied to a FMN, which was performed by means of simulation, using the Network Simulator 3 (ns-3). The routing protocols under study, including AODV, OLSR, RedeFINE, and static routing, are characterized in terms of throughput, Packet Delivery Ratio (PDR), and delay. Results show the superior performance of RedeFINE, namely concerning aggregate throughput and PDR, with gains up to 65%. Regarding end-to-end delay, AODV and OLSR provide better results.

***Index Terms*—Unmanned Aerial Vehicles, Flying Multi-hop Networks, Routing protocols.**

I. Introduction

In the last years, the usage of Unmanned Aerial Vehicles has become popular in several applications, including environmental monitoring, border surveillance, search and rescue, logistics, and payload transport [1]. The capability to operate anywhere, their mobility and hovering capabilities, an their growing payload make them viable platforms to perform such tasks with effectiveness. Recently, the advent of small and low-cost UAVs paved the way to the usage of swarms of UAVs, which cooperate between themselves and perform missions more efficiently than a standalone UAV. However, cooperation requires reliable and stable communications between the UAVs, giving rise to the concept of Flying Multi-hop Networks (FMNs). FMNs are a particular case of Mobile Ad-Hoc Networks (MANETs) and Vehicular Ad-Hoc Networks (VANETs), since they introduce new challenges, namely concerning network routing. In fact, FMNs have a highly 3-dimensional dynamic behaviour due to the highly mobility capability of the UAVs, which introduce frequent changes in the network topology and in the quality of the radio links. More recently, FMNs are also being used to provide Internet access in remote areas where a network infrastructure does not exist, and to enhance the capacity of existing networks in temporary crowded events, such as fun parks, music festivals, and fairs, as depicted in Fig. 1 [2]. These scenarios bring up even more significant routing challenges, in order to attend the users' traffic demands, including when the UAVs are moving.

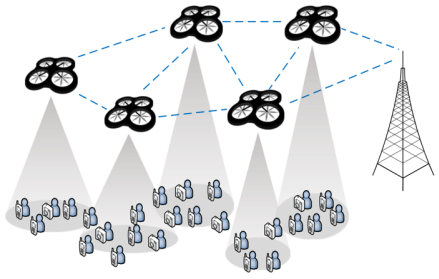


Fig. 1. FMN providing Internet connectivity to the terminals on the ground [3].

Nevertheless, existing routing protocols for ad-hoc networks have been designed for MANETs and VANETs and their application in FMNs remains not well characterized.

Motivated by the need of developing new routing solutions to meet the challenges introduced by emerging scenarios using UAVs, in [4] the authors propose RedeFINE, a centralized routing solution, based on the principles of Software-Defined Networking (SDN), especially targeted for FMNs. It takes advantage of knowing in advance the future trajectories of the UAVs, which is provided by a Central Station in charge of make such planning, to calculate in advance the routing tables and the instants they shall be updated in the UAVs. This routing solution does not use control packets for neighbor discovery and link sensing, allowing to maximize the bandwidth available for data traffic. In addition, it enables uniterruptible communications between the UAVs, by updating the routing tables in the UAVs before link failures occur. However, this routing solution was only evaluated under two network scenarios composed of UAVs with reduced movements variability.

The main contributions of this paper are:

- Performance evaluation of a set of state of the art routing protocols applied to a FMN providing Internet access in a crowded event, including:
  - Two distributed routing protocols representative of the reactive and proactive routing paradigms: Ad Hoc On-Demand Distance Vector (AODV) and Optimized Link State Routing Protocol (OLSR), respectively.

  - RedeFINE, which is representative of centralized routing, following the principles of SDN.
  - Static routing, for which the forwarding tables are maintained throughout the whole time.
- Validation of RedeFINE under random topologies, with UAVs moving at higher speeds, which are aspects left for future work in [4].

The study was performed by means of simulation, using the Network Simulator 3 (ns-3). The routing protocols under study are characterized in terms of throughput, Packet Delivery Ratio (PDR), and delay. Results show the superior performance of RedeFINE, namely concerning aggregate throughput and PDR, while AODV and OLSR provide better results for end-to-end delay.

The rest of the paper is organized as follows. Section II presents the state of the art on routing solutions for FMNs. Section III defines the system model and formulates the problem. Section IV presents the concept of the routing protocols under study. Section V addresses their performance evaluation, including the simulation setup, the simulation scenarios, the performance metrics studied, and the simulation results. Section VI discusses the simulation results and the pros and cons of the evaluated routing protocols. Finally, Section VII points out the main conclusions and directions for future work.

II. State of the Art

Current routing solutions for FMNs are based on the protocols designed for Mobile Ad Hoc Networks (MANETs) and Vehicular Ad Hoc Networks (VANETs). They may be divided into two categories: single-hop routing and multi-hop routing.

In single-hop routing, a pre-defined routing configuration, which does not need to be updated, is used to forward the packets. In the context of UAVs' networks, the UAVs act as packet carriers, by loading data from ground nodes and carrying it to the destination. This routing solution is best suited for delay-tolerant networks, enabling high throughput and PDR, since interference and medium access contention are avoided when UAVs are used to transport the packets. However, since mechanisms for detection of invalid routes are typically not available, single-hop routing is manly suitable for networks with fixed topology [5, 6].

In multi-hop routing, the packets are forwarded hop by hop, along paths composed of multiple nodes. Actually, multi-hop routing protocols are the most used, so they deserve special focus in what follows. Multi-hop routing solutions are mainly divided into topology-based and position-based routing protocols, based on the metrics employed to select the next-hop nodes.

Topology-based routing protocols use metrics such as hop-count and the quality of the communications links to perform the selection of the next-hop nodes; they may be classified as proactive, reactive and hybrid [5]. Proactive routing protocols aim at maintaining the routing tables of all nodes always up-to-date, so that the paths can be selected immediately, thus reducing the waiting time to forward packets. This is achieved by broadcasting 1) HELLO packets, for neighbor discovery, and 2) topology control packets, for routes announcement. However, proactive routing protocols need a large amount of control packets, and are not suitable for high-mobility networks [4, 5]. In an opposite way, reactive routing protocols perform the routing decisions on-demand when a packet needs to be sent. Therefore, they reduce the overhead of the control packets of the proactive routing protocols and the power consumption, however introduce high delay to define the paths. Reactive routing protocols are best-suited for highly mobility scenarios with low traffic-load [4, 5, 6]. Finally, hybrid routing protocols combine the advantages of proactive and reactive routing protocols. They aim at overcoming the overhead introduced by the proactive routing protocols and the long delay of the reactive routing protocols. Hybrid routing protocols are appropriate for large networks, which must be divided into a set of zones; intra-zone routing is performed using the proactive approach, and inter-zone routing is performed using the reactive approach [4, 5, 6, 7].

Position-based protocols rely on the geographic location information to perform the routing decisions. They aim at giving response to the fast network topology changes of highly mobility networks, which is a drawback of the proactive routing protocols, and minimize the high-delay inherent to the routing discovery process of the reactive routing protocols. However, position-based protocols require accurate location information of the communications nodes in short time intervals, which is only possible with expensive hardware, and are not proper for sparse networks [4, 5, 6, 7].

The previous solution rely on the distributed routing paradigm. However, recently, the Software-Defined Networking (SDN) paradigm, which was successfully applied to wired networks, has been introduced in wireless networks, giving rise to the concept Software-Defined Wireless Networking (SDWN). Inspired by this paradigm, in [4] the authors introduced RedeFINE, a centralized routing solution especially targeted for FMNs. RedeFINE takes advantage of a holistic and centralized view of the network to perform the routing decisions, promising high-capacity and uninterruptible communications between UAVs. However, RedeFINE was only validated for two scenarios, with reduced number of UAVs generating traffic and low movements variability. To the best of our knowledge, a complete performance study of this solution has not yet been presented.

In the literature, few works study the applicability of existing ad-hoc routing protocols in FMNs. The most relevant ones are presented in [8, 9, 10, 11], however they are tailored for low-traffic applications. This paper considers a FMN able to provide Internet connectivity to ground users, considering high-capacity requirements, and including a centralized routing approach: RedeFINE.

III. System Model

The network used in this study, hereafter named FMN, consists of N UAVs of two types: Flying Mesh Access Points (FMAPs), which provide Internet connectivity to ground users,

and a Gateway (GW) UAV, which forwards the traffic to the Internet. We model the FMN at time instant $t_k = k \cdot \Delta t, k \in N_0$ and $\Delta t \in \mathbb{R}$ as a directed graph $G(t_k) = (V, E(t_k))$, where $V \in \{1, ..., N\}$ is the set of UAVs and $E(t_k) \subseteq V \times V$ is the set of directional communications links between any two UAVs i and j, at t_k, where $i, j \in V$. The channel between any two UAVs is modeled by Friis, since a strong Line-of-Sight (LoS) component characterizes the communications links between UAVs flying dozens of meters above the ground. For RedeFINE, the directional wireless communications link $(i, j)_{t_k}$ exists if and only if the power $P_{R_i}(t_k)$ received by UAV i at time t_k divided by the noise power N_i satisfies (1), that is, if the Signal-to-Noise Ratio (SNR) is higher than threshold S. The received power at UAV i, $P_{R_i}(t_k)$, results from the Friis path loss model defined in (2), where $P_{T_j}(t_k)$ describes the power transmitted by UAV j at time t_k, $\lambda_{i,j}$ denotes the link wavelength, and $d_{i,j}(t_k)$ expresses the Euclidean distance between UAV i and UAV j at time t_k. A path is defined as a set of adjacent links connecting UAV i to the GW UAV; multiple paths may be available for UAV i at t_k, but only one of them is used.

$$\frac{P_{R_i}(t_k)}{N_i} > S \tag{1}$$

$$\frac{P_{R_i}(t_k)}{P_{T_j}(t_k)} = \left(\frac{\lambda_{i,j}}{4\pi \times d_{i,j}(t_k)}\right)^2 \tag{2}$$

Considering the throughput $R_i(t_k)$, in bit/s, as the bitrate of the flow from UAV i received at the GW UAV at time t_k, and N UAVs generating traffic towards the GW UAV, each routing protocol in study must maximize at any time instant t_k the amount of bits received by the GW UAV during time interval Δt. As such, our objective function can be defined as:

$$\max \sum_{i=0}^{N-1} R_i(t_k) \times \Delta t \tag{3}$$

The factors influencing $R_i(t_k)$ include the capacity of the path used by UAV i, which should be limited by the link in the path having the smallest capacity, the number of flows traversing the links, medium access protocol behaviour, and interference between the communications nodes.

IV. Routing Protocols

In this section, the routing protocols under study are presented. From the distributed routing protocols, Ad Hoc On-Demand Distance Vector (AODV) is representative of the reactive routing paradigm, and Optimized Link State Routing Protocol (OLSR) is representative of the proactive routing paradigm. RedeFINE is representative of the centralized routing paradigm. Static routing is also considered.

A. Ad Hoc On-Demand Distance Vector (AODV)

AODV [12] discovers the routes on-demand basis, and they are maintained as long as they are required. AODV uses a sequence number that increases each time it detects a change in the topology formed by its neighbour nodes. This sequence number ensures that the most recent routes are selected, and avoids loops. AODV employs an important mechanism named Route Request Packet (RREQ). RREQ packets are broadcasted to the network to discover the routes to a predefined IP address. When a node has a valid connection to the required destination, it responds with a Route Reply Packet (RREP), which is used by the source of the initial RREQ packet to build the route to the destination.

B. Optimized Link State Routing Protocol (OLSR)

In OLSR [13], all nodes have complete information about their neighboring nodes. In order to reduce the overhead that characterizes proactive routing protocols, OLSR uses the concept of Multi-Point Relays (MPR): a node (selector), independently, chooses a minimal subset of its one-hop neighbors, known as MPRs, to reach all its two-hop neighbors. Two different types of messages are broadcasted in the network: HELLO messages, which are used to discover information about link status, and Topology Control (TC) messages, which are used to exchange information regarding network changes. When a nodes sends/forwards a TC message, only its MPRs forward the message, reducing retransmissions.

C. RedeFINE

RedeFINE was designed to take advantage of the centralized view of the network provided by a Central Station (CS). The CS is in charge of defining the future positions of the UAVs, so that they can meet the traffic demands of the users on the ground. RedeFINE selects periodically the best path for each UAV, which is defined as the path with the shortest Euclidean distance. For that, it uses the Dijkstra algorithm. RedeFINE assumes a strong Line-of-Sight (LoS) between the UAVs, which is characteristic of UAVs flying dozens of meters above the ground.

D. Static routing

In static routing, which is the simplest routing configuration, the forwarding tables defined by RedeFINE for the initial instants are maintained during the whole time.

V. Performance Evaluation

The methodology used in the performance evaluation is explored in this section, including the simulation setup, the simulated scenarios, and the performance metrics.

A. Simulation Setup

The performance evaluation was carried out using the ns-3 simulator. The FMN under study was composed of 1 GW UAV and 20 FMAPs. In each UAV, a Network Interface Card (NIC) was configured in Ad Hoc mode, using the IEEE 802.11ac standard at channel 50, which allows 160 MHz channel bandwidth. The data rate was defined by the *IdealWifiManager* mechanism. The wireless links between the UAVs were modeled by Friis path loss; for RedeFINE, only links with SNR above 5 dB were considered. The transmission power of the NICs was set to 0 dBm. One IEEE 802.11ac spatial stream was

TABLE I
SUMMARY OF THE NS-3 SIMULATION PARAMETERS.

Simulation time	(30 s initialization +) 130 s
Wi-Fi standard	IEEE 802.11ac
Channel	50 (5250 MHz)
Channel bandwidth	160 MHz
Guard Interval	800 ns
TX power	0 dBm
Propagation delay model	Constant speed
Propagation loss model	Friis path loss
Remote station manager	IdealWifiManager
Wi-Fi mode	Ad Hoc
Mobility model	Waypoint mobility
Traffic type	UDP Poisson
Packet size	1400 bytes

used for all inter-UAV wireless links. With 1 spatial stream, the data rate corresponding to the maximum Modulation and Coding Scheme (MCS) index is 780 Mbit/s, considering 800 ns Guard Interval. Taking into account the dimensions of the simulated scenarios, we assume an average number of 2 hops between the FMAPs generating traffic and the GW UAV; this results in $\frac{\frac{780}{10}}{2}\mathrm{Mbit/s}$ for the maximum achievable data rate per flow, considering 10 FMAPs generating traffic. Based on that, the maximum offered load for each scenario was set to 25% and 75% of the maximum achievable data rate per flow. The traffic generated was UDP with arrival process modeled as Poisson, for a constant packet size of 1400 bytes; the traffic generation was only triggered after 30 s of simulation, in order to ensure a stable state.

A summary of the ns-3 simulation parameters used is presented in Table I.

B. Simulation Scenarios

Five scenarios, in which UAVs were moving according to the Random Waypoint Mobility (RWM) model, were generated. Under the RWM model, each UAV chooses a random destination and a speed uniformly distributed between a minimum and a maximum values. Then, the UAV moves to the chosen destination at the selected speed; upon arrival, the UAV stops for a specified period of time, and repeats the process for a new destination and speed [14].

Since RedeFINE solution relies on knowing in advance the movements of the UAVs, instead of generating the random movements in real-time during the ns-3 simulation, we used BonnMotion [15], which is a mobility scenario generation tool. BonnMotion was set to create Random Waypoint 3D movements for 21 nodes (20 FMAPs and 1 GW UAV) within a box of dimensions 80 m×80 m×25 m, during 160 s, considering a velocity between 0.5 m/s and 3 m/s for the UAVs. These scenarios were used to calculate in advance the forwarding tables and the instants they shall be updated in the UAVs. The generated scenarios, as well as the forwarding tables that are calculated in advance by RedeFINE, were finally imported to the ns-3 simulator with a sampling period of 1 s. To employ mobility to the UAVs, based on the generated scenarios, the *WaypointMobilityModel* model of ns-3 was used.

C. Performance Metrics

The performance evaluation of each routing protocol presented in Section IV considers three performance metrics:

- **Aggregate throughput** – The average number of bits received per second by the GW UAV.
- **Packet Delivery Ratio (PDR)** – The number of packets received by the GW UAV divided by the number of packets generated by the FMAPs, measured at each second.
- **End-to-end delay** – The time taken by the packets to reach the application layer of the GW UAV since the instant they were generated, measured at each second, including queuing, transmission, and propagation delays.

Since the default implementations of AODV and OLSR in the latest release of ns-3 do not employ a link quality-based routing metric, we used alternative implementations for both protocols, publicly available in [16] and [17], respectively, in order to enable a fair comparison with RedeFINE. They use the Expected Transmission Count (ETX) [18], which represents the predicted number of transmissions required to send a packet over a link, including retransmissions.

D. Simulation Results

The simulation results are presented in this subsection. The results were obtained after 20 simulation runs using different seeds. The results are represented by means of the Cumulative Distribution Function (CDF) for the mean end-to-end delay, and by the Complementary CDF (CCDF) for the mean aggregate throughput and PDR, considering five random scenarios, as stated in Subsection V-B. The CDF $F(x)$ expresses the percentage of simulation time for which the mean end-to-end delay was lower or equal to x, while the CCDF $F'(x)$ expresses the percentage of simulation time for which the mean mean aggregate throughput or PDR was higher than x. The mean aggregate throughput versus the simulation time is also presented.

The CCDF representation for the mean aggregate throughput in the GW UAV (c.f. Fig. 2), considering different packet generation rates, shows that RedeFINE enables higher throughput than its state of the art counterparts. While in static routing the path to the GW UAV becomes unusable, due to the movements of the UAVs, in AODV a considerable amount of time is spent finding the paths to the GW UAV, which decreases the throughput substantially, as depicted in Fig. 2. In OLSR, the routes are changed in short time periods, hence many packets are dropped during these updates. In addition, in AODV and OLSR, the ETX link estimation is based on small packets; as consequence, the link loss rate may be underestimated, especially if high data rates are used, as occurs in IEEE 802.11ac. The superior performance of RedeFINE is justified by the selection of stable paths, with minimal length, composed of links with high SNR, high capacity, and low delay. Considering the areas under the CCDF curves for the mean aggregate throughput (c.f. Fig. 2a and Fig. 2b), which give the total amount of bits in the GW UAV, we conclude

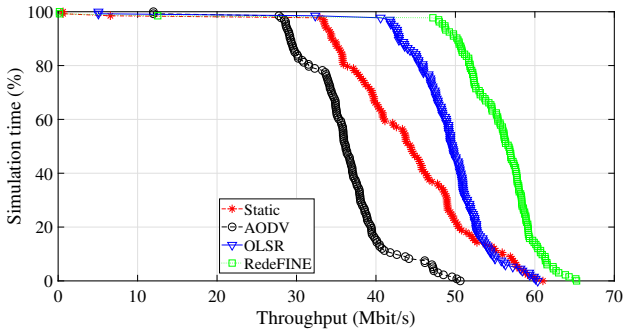
that RedeFINE increases the number of bits received in the GW UAV up to 13%, 30%, and 65% in relation to OLSR, static routing, and AODV, respectively. The CCDF for the PDR (c.f Fig. 3) has the same pattern as the CCDF for the mean aggregate throughput. Overall, the PDR for $\lambda \approx 30\,\mathrm{Mbit/s}$ is lower than for $\lambda \approx 10\,\mathrm{Mbit/s}$. This is justified by the higher level of congestion that the FMN is subject in the former case. Nevertheless, RedeFINE enables higher PDR than its state of the art counterparts for both cases. Regarding the end-to-end delay, OLSR provides a superior performance with respect to the remaining routing protocols for $\lambda \approx 10\,\mathrm{Mbit/s}$ (c.f. Fig. 4), while for $\lambda \approx 30\,\mathrm{Mbit/s}$ the end-to-end delay provided by of OLSR is matched by AODV. The worse performance of RedeFINE for this metric is justified by its higher PDR. Since more packets are delivered, the FMN is subject to a higher level of congestion; hence, the packets are held in the transmission queue longer, which increases the end-to-end delay. In turn, the high end-to-end delay in static routing is justified by unstable links with low SNR and low capacity, due to the inability to update the routing tables over time.

VI. Discussion

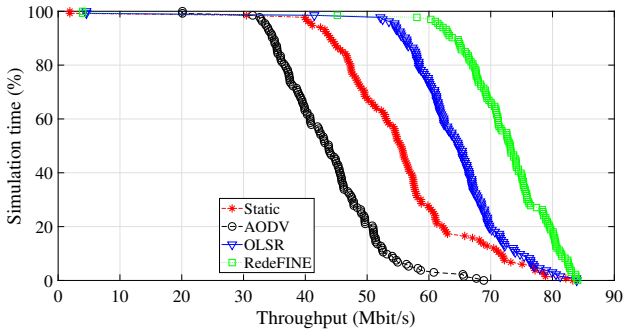
The simulation results show that RedeFINE improves the performance of a FMN, especially regarding throughput and PDR. RedeFINE provides a superior performance than AODV and OLSR, even when they use ETX, which is a link quality-based metric. RedeFINE enables the selection of stable paths with minimal length, formed by links with high SNR, high capacity, and low delay. In addition, RedeFINE is more appropriate for scenarios where UAVs move at high speeds, since it reacts to the topology changes in advance, contrary to the distributed routing protocols. To achieve similar results using a distributed routing protocol, an option would be to reduce the interval between HELLO packets, so that the routing tables could be updated with the desired responsiveness, in order to support high mobility scenarios. However, the higher number of HELLO packets would reduce the bandwidth available for data traffic. RedeFINE does not use control packets for neighbor discovery and link sensing, and the routing tables are computed centrally, which allows to reduce the computational power on board of the UAVs; the control traffic is only formed by the forwarding tables sent from the CS to the UAVs. Nevertheless, the stability and synchronism of the whole FMN when RedeFINE is used must be studied in future works. As final remark, we denote that the evaluated protocols are not suitable for highly dense networks, where the UAVs can be forwarding traffic from multiple sources; hence, this aspect must be addressed in future routing protocols developed for FMNs.

VII. Conclusions

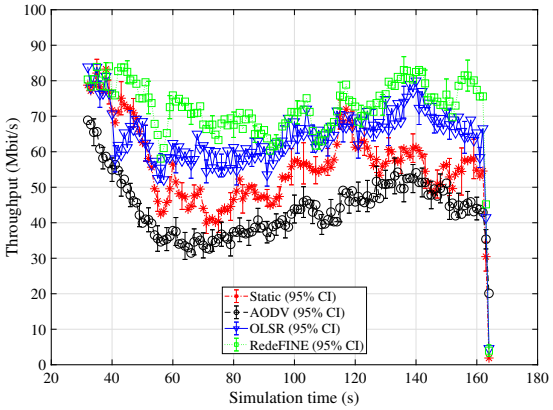
We presented a performance evaluation of routing protocols applied to a FMN. Two state of the art routing protocols representative of the reactive and proactive routing paradigms, AODV and OLSR, a centralized routing protocol, RedeFINE,



(a) Throughput CCDF for $\lambda \approx 10\,\mathrm{Mbit/s}$.



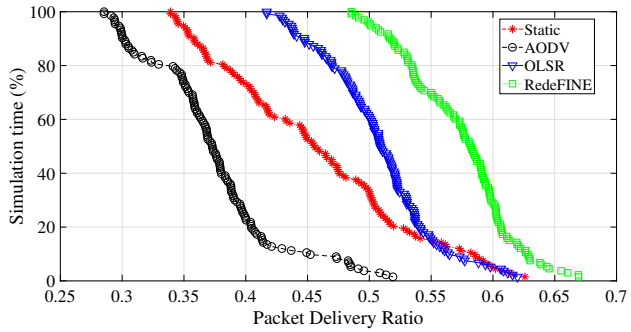
(b) Throughput CCDF for $\lambda \approx 30\,\mathrm{Mbit/s}$.



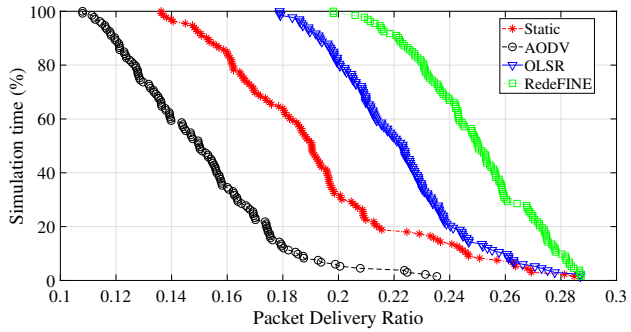
(c) Throughput versus simulation time for $\lambda \approx 30\,\mathrm{Mbit/s}$.

Fig. 2. Throughput results, considering 10 FMAPs generating traffic.

and static routing, for which the forwarding tables are maintained during the whole time, were evaluated. Based on simulation results, we demonstrated the superior performance of RedeFINE, regarding throughput and PDR. The results showed gains up to 65%. Regarding end-to-end delay, the results demonstrated a superior performance of OLSR and AODV. RedeFINE did not achieve the values obtained by OLSR and AODV due to its higher PDR, which increases network congestion and the time packets are held in the transmission queue. As future work, the stability and synchronism of the FMN when RedeFINE is used must be studied. In addition, we pointed out as a research challenge the development or

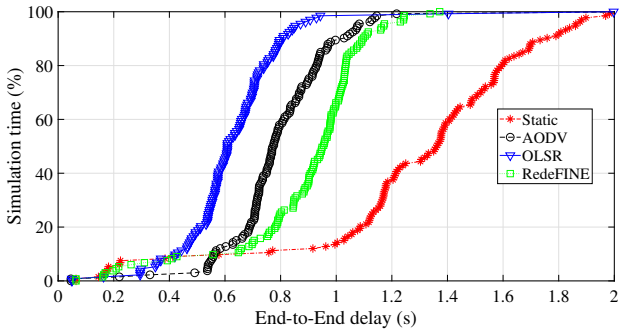


(a) PDR CCDF for $\lambda \approx 10\,\mathrm{Mbit/s}$.

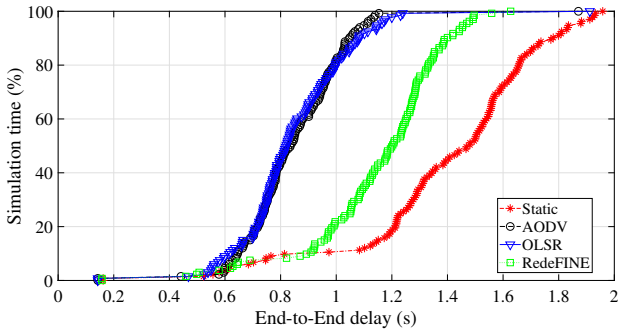


(b) PDR CCDF for $\lambda \approx 30\,\mathrm{Mbit/s}$.

Fig. 3. PDR results, considering 10 FMAPs generating traffic.



(a) End-to-end delay CDF for $\lambda \approx 10\,\mathrm{Mbit/s}$.



(b) End-to-end delay CDF for $\lambda \approx 30\,\mathrm{Mbit/s}$.

Fig. 4. End-to-end delay results, considering 10 FMAPs generating traffic.

improvement of a routing protocol for FMNs able to take into account the traffic load of the UAVs, in order to be suitable for highly dense FMNs.

References

[1] Samira Hayat, Evşen Yanmaz, and Raheeb Muzaffar. "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint". In: *IEEE Communications Surveys and Tutorials* 18.4 (2016), pp. 2624–2661. ISSN: 1553877X. DOI: 10.1109/COMST.2016.2560343.

[2] Shams ur Rahman et al. "Positioning of UAVs for throughput maximization in software-defined disaster area UAV communication networks". In: *Journal of Communications and Networks* 20.5 (2018), pp. 452–463. ISSN: 1229-2370. DOI: 10.1109/JCN.2018.000070. URL: https://ieeexplore.ieee.org/document/8533581/.

[3] INESC TEC. *WISE*. http://wise.inesctec.pt/.

[4] A. Coelho et al. "RedeFINE : Centralized Routing for High-capacity Multi-hop Flying Networks". In: *2018 IEEE International Conference on Wireless and Mobile Computing* 2018-Octob (2018), pp. 1–6.

[5] J Jiang and G Han. "Routing Protocols for Unmanned Aerial Vehicles". In: *IEEE Communications Magazine* 56.1 (Jan. 2018), pp. 58–63. ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1700326.

[6] Ozgur Koray Sahingoz. "Networking models in flying ad-hoc networks (FANETs): Concepts and challenges". In: *Journal of Intelligent & Robotic Systems* 74.1-2 (2014), pp. 513–527.

[7] Md Hasan Tareque, Md Shohrab Hossain, and Mohammed Atiquzzaman. "On the routing in flying ad hoc networks". In: *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*. IEEE. 2015, pp. 1–9.

[8] K. Singh and Anil Kumar Verma. "Experimental analysis of AODV, DSDV and OLSR routing protocol for flying adhoc networks (FANETs)". In: *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. Mar. 2015, pp. 1–4. DOI: 10.1109/ICECCT.2015.7226085.

[9] Danil S Vasiliev, Daniil S Meitis, and Albert Abilov. "Simulation-based comparison of AODV, OLSR and HWMP protocols for flying Ad Hoc networks". In: *International Conference on Next Generation Wired/Wireless Networking*. Springer. 2014, pp. 245–252.

[10] Artur Carvalho Zucchi and Regina Melo Silveira. "Performance Analysis of Routing Protocol for Ad Hoc UAV Network". In: *Proceedings of the 10th Latin America Networking Conference*. ACM. 2018, pp. 73–80.

[11] Anand Nayyar. "Flying Adhoc Network (FANETs): Simulation Based Performance Comparison of Routing Protocols: AODV, DSDV, DSR, OLSR, AOMDV and HWMP". In: Aug. 2018, pp. 1–9. DOI: 10.1109/ICABCD.2018.8465130.

[12] Charles Perkins, Elizabeth Belding-Royer, and Samir Das. *Ad hoc on-demand distance vector (AODV) routing*. Tech. rep. 2003.

[13] Thomas Clausen and Philippe Jacquet. *Optimized link state routing protocol (OLSR)*. Tech. rep. 2003.

[14] Tracy Camp, Jeff Boleng, and Vanessa Davies. "A survey of mobility models for ad hoc network research". In: *Wireless Communications and Mobile Computing* 2.5 (2002), pp. 483–502. ISSN: 15308669. DOI: 10.1002/wcm.72. arXiv: arXiv:1011.1669v3.

[15] Nils Aschenbruck et al. "BonnMotion: a mobility scenario generation and analysis tool". In: *Proceedings of the 3rd international ICST conference on simulation tools and techniques*. ICST (Institute for Computer Sciences, Social-Informatics and ... 2010, p. 51.

[16] *GitHub - neje/ns3-aodv-etx*. https://github.com/neje/ns3-aodv-etx. (Accessed on 01/13/2019). Apr. 2018.

[17] *GitHub - igorcompuff/ns-3.26*. https://github.com/igorcompuff/ns-3.26. (Accessed on 01/13/2019). May 2017.

[18] Douglas S. J. De Couto et al. "A High-throughput Path Metric for Multi-hop Wireless Routing". In: *Wirel. Netw.* 11.4 (July 2005), pp. 419–434. ISSN: 1022-0038. DOI: 10.1007/s11276-005-1766-z. URL: http://dx.doi.org/10.1007/s11276-005-1766-z.

A Survey on Device-to-Device Communication in 5G Wireless Networks

Amir Hossein Farzamiyan
dept. Electrical Engineering
University of Porto
Porto, Portugal
up201809136@fe.up.pt

***Abstract*—The Device-to-Device (D2D) communication model in 5G networks provides a useful infrastructure to enable different applications. D2D communication, with use of cellular or ad-hoc links, improve the spectrum utilization, system throughput, and energy efficiency of the network thereby preparing the ability for the user equipment to start communications with each other in proximity. The purpose of this paper is preparing a survey based on the D2D communication and review the available literature that in a widespread way research about the D2D paradigm, different application scenarios, and use cases. Moreover, new suspicion in this area that leads to identifying open research problems of D2D communications in cellular networks.**

***Index Terms*—Device-to-device communication, D2D, 5G networks, Cellular network, Survey**

I. INTRODUCTION

Cellular communication will face with the fifth generation (5G) soon. In order to successfully handle all the demands of the subscribers for higher data rates and support several applications, make 4G systems be replaced by 5G. Considering the current 4G technologies cannot fulfill the huge gap between the actual communication performances and the forthcoming user expectations, Third Generation Partnership Project (3GPP) has been developing an enhanced Long-Term Evolution (LTE) radio interface called LTE-Advanced (LTE-A). LTE-A radio interface is designed with advanced communication techniques such as carrier aggregation, massive Multiple-Input Multiple-Output (MIMO), low-power nodes, as well as D2D communication, which are expected to dramatically improve the current cellular technologies (4G) in terms of system capacity, coverage, peak rates, throughput, latency, user experience, etc. [1].

5G is the result of using various technologies like mm-Wave communication, Massive MIMO, and Cognitive Radio Networks (CRNs) [1]. 5G, despite of the first four generations of cellular networks that completely dependent upon the base station (BS), is heading towards device-centric approach, which means that network setup is managed by the devices themselves.

A rigorous growth exists in networks traffic over the years and will continuously increase in the following years, as depicted in Fig. 1. This results in overloading at the base station (BS). Due to this mounting load on the base station (BS), there is an increase in the demand for power offloaded from the base station and here D2D communication plays a crucial role. Since D2D communication allows devices to communicate with each other without traversing the base station, load on the base station is highly reduced.

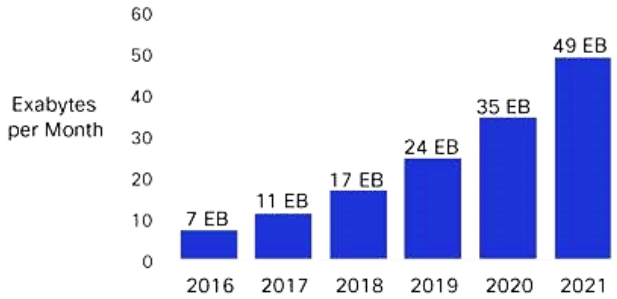


Fig. 1. Cisco forecasts 49 Exabytes per month of mobile data traffic by 2021 [2]

Some have speculated that Wi-Fi offload will be less relevant after 4G networks are in place because of the faster speeds and more abundant bandwidth. However, 4G networks have attracted high-usage devices such as advanced smartphones and tablets, and now 4G plans are subject to data caps similar to 3G plans. For these reasons, Wi-Fi offload is higher on 4G networks than on lower-speed networks, now and in the future according to Cisco projections. The amount of traffic offloaded from 4G was 63 percent at the end of 2016, and it will be 66 percent by 2021 (Fig. 2) [2].

The amount of traffic offloaded from 3G will be 55 percent by 2021, and the amount of traffic offloaded from 2G will be 69 percent. As 5G is being introduced, plans will be generous with data caps and speeds will be high enough to encourage traffic to stay on the mobile network instead of being offloaded, so the offload percentage will be less than 50 percent. As the 5G network matures, we may see higher offload rates thereby D2D communication as one more mechanism for network offloading will become more applicable.

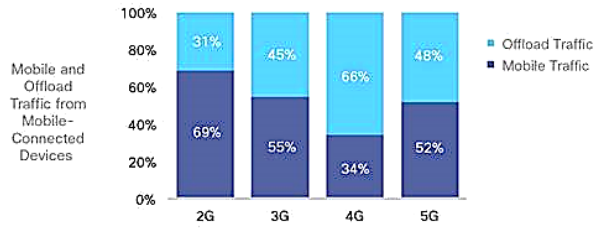


Fig. 2. Mobile data traffic and offload traffic [2]

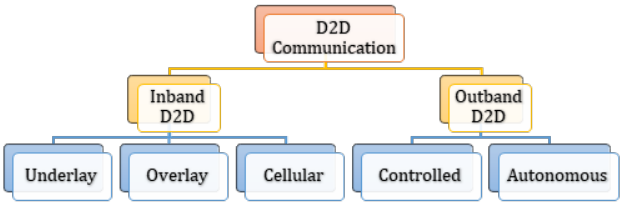


Fig. 4. D2D Communication Category

II. OVERVIEW OF DEVICE-TO-DEVICE (D2D) COMMUNICATION

D2D communication is being considered as an essential component of the 5G networks. Communication features such as system capacity, throughput, spectral efficiency, and latency are expected to improve with the help of applying D2D communication technique citec3,c4. In [5], the evolutionary development of cellular communication generations has been given. An overview of the services supported by the generations of cellular communications is shown in Fig. 3.

While working on the D2D technology, some challenging issues like interference management, radio resource allocation, procedures management, and communication session setup appear in the cellular network and are reported in the recent literature [6]–[9]. A. Asadi et. al. [3], have been proposed several taxonomies of possible D2D architectures. In particular, D2D communications separate into two main categories, in-band, and out-band. The first category uses radio spectrum that occurs on the cellular spectrum while the other use unlicensed spectrum.

When the communication is in the unlicensed spectrum, the coordination between radio interfaces is either controlled autonomously by mobile terminations (MTs) or by BS (i.e., controlled). Interference mitigation between cellular and D2D communications is the main challenging issues on in-band D2D communications and several research proposals focus on the study of this problem [10], [11]. Concerning out-band D2D communications, the research focuses on inter-technology architectural design and power consumption [12]. All these proposals point out the potentialities of the different approaches in terms of energy consumption and of bandwidth resources.

One of the most significant challenges in in-band D2D communication is how to allocate spectrum for such type of communication. The classification of D2D communication as resource allocation is depicted in Fig. 4. Up to now, there are

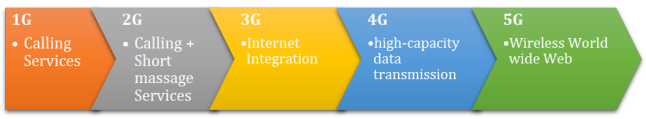


Fig. 3. Generation of cellular communication

three resource allocation modes for reusing licensed spectrum resources [13]:

- **Underlay Mode:** D2D pairs and cellular user equipment (UEs) share the same spectrum resources, which has the advantage of achieving the best spectrum efficiency. It is noticed that in underlay mode, one of the key issues is to effectively control the D2D-to-cellular and cellular-to-D2D interference.
- **Overlay Mode:** Dedicated frequency resources are allocated for D2D communications, and the remaining part is allocated for cellular communications. In such mode, there is no interference issue between D2D and cellular communications. One research focus is how to optimize the resource allocation ratio.
- **Cellular Mode:** Instead of communicating directly with each other, D2D UEs communicate with the eNB acting as an intermediate relay, which is the same as the traditional cellular system.

III. APPLICATION SCENARIOS AND ADVANTAGES OF D2D COMMUNICATIONS

In this section, first, different D2D communications application scenario is explored then discuss more the advantages of D2D communications while comparing with similar networks.

A. Use cases and usage scenarios

Various use cases and application scenarios of D2D communications have been proposed. As shown in Fig. 5 and according to the participation of cellular base stations or core networks, D2D communications scenarios categorize into three representative types.

1) In-Coverage: D2D communications between two user devices are fully controlled by the network infrastructure of operators, such as BS or core networks. In this scenario, all user devices are located in the coverage of cellular networks. The operator manages the shared cellular licensed spectrum between the D2D links and normal cellular connections. Typical use cases of this scenario not limited to local traffic offloading from the core networks and operator controlled local data services, such as local content sharing, gaming, and Machine-to-Machine (M2M) communications.

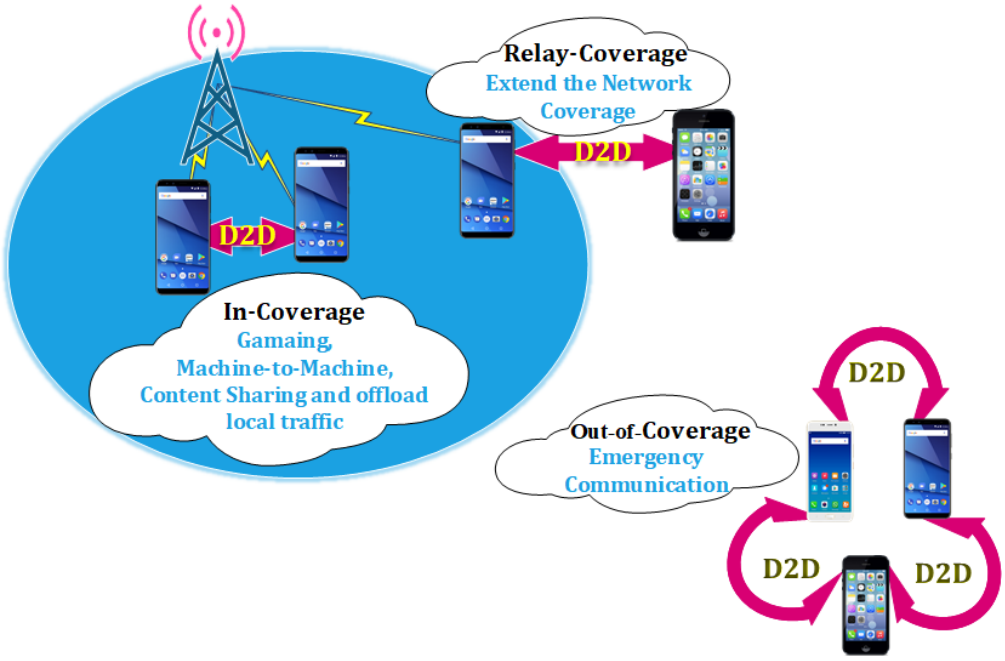


Fig. 5. D2D communication application scenario

2) Relay-Coverage: In this scenario, D2D communications can improve network service quality at the edge of network coverage by extending the coverage of cellular networks. User devices that are out of Base Station (BS) coverage can use other covered devices as a data communication relay and by means of them communicate with the core network (BS). Like the previous scenario, the operator fully controls the connection establishment, resource allocation for both User-to-BS connections and D2D (User-to-User) connections and D2D link used the shared cellular licensed spectrum.

3) Out-of-Coverage: This D2D communication scenario looks similar to MANETs. Out-of-Coverage scenario serves as the important component for emergency communication services, (e.g., national security, disaster relief, and public safety communications) [14] [15]. In an urgent situation where the cellular infrastructure has been severely damaged, caused by a flood, storm, fire etc., D2D user devices, without the assistance of any operators, can establish connections and start D2D communications with each other in proximity.

D2D communications are expected to be an underlying network of LTE-Advanced (LTE-A). In order to introduce the D2D communications into existing LTE Networks and make them compatible with LTE-A, the 3rd Generation Partnership Project (3GPP) proposed ProSe (i.e., D2D communications) system architecture under the framework of LTE Networks [16]. Vehicle-to-Vehicle (V2V) applications for safety and infotainment are based on IEEE 802.11p [17].

Internet-of-Things (IoT) is defined as the interconnection via the internet of computing devices embedded in everyday objects, enabling them to send and receive data. M2M communication is a form of data communication that involves one or more entities that do not necessarily require human interaction or intervention in the process of communication [18]. M2M communication is also named as Machine Type Communication (MTC) in 3GPP. This type of communication could be carried over mobile networks (e.g., LTE or LTE-A) and regarded as an underlaying technology on IoT. D2D communications can apply for M2M communications in the IoT, which means that under the supervision and control of core networks, like Base Station or M2M server, enable intelligent machines to interchange data, communicate directly with each other, and consequently improve network performance, lower power consumption, and reduce transmission delay due to offload the core network local traffic.

B. D2D Communications Advantages

There are lots of study in D2D communications technology to improve the services quality and facilitation. In summary, these services put in three major categories described below.

1) Emergency communications: [19]–[22] In the case of natural disasters like hurricanes, earthquakes etc., the traditional communication network may not work due to the damage caused. Ad-hoc network can be established via D2D which could be used for such communication in such situations.

2) IoT Sweetening: [23], [24] By combining D2D with IoT, a truly interconnected wireless network will be created. Example of D2D-based IoT enhancement is the improvement in Internet of Vehicles (IoV) when two vehicles running at high speeds, a vehicle can warn nearby vehicles in the D2D mode about speed or other information.

3) Local Services: [25], [26] In local service, user data is directly transmitted between the terminals and doesnt involve

network side, e.g. social media apps, which are based on proximity service.

Although D2D communications on many aspects similar to MANETs [27] but some differences are easy to perceive. First, D2D communications can work on licensed or unlicensed spectrums in different scenarios while MANETs work independently on unlicensed spectrums. Interference is the main problem in MANETs due to difficult spectrum control on unlicensed spectrums whereas in D2D control of core networks on efficient spectrum resources consumption, minimize the interference between links occurred. However, in the Out-of-Coverage scenario, the D2D communications occur either on unlicensed spectrums like MANETs or in case such as Public Safety Network occur on licensed spectrums [28].

Second, in D2D communications, operations such as resource allocation, node discovery, route search and security management can be performed through the core networks and D2D nodes cooperation or controlled by core networks. While in MANETs each node performs the above-mentioned operations autonomously.

Finally, the distinct difference between MANETs and D2D communications is the routing patterns. D2D communications mainly put single hop communications into services while the leading and troublesome challenges in MANETs that need to consider are the issues of multi-hop routing. It should be considered that in Out-of-Coverage scenario, D2D communications like MANETs faces the same issues in multi-hop routing.

IV. CONCLUSION

This survey showed that Device-to-Device (D2D) communication in cellular networks is an emerging wireless technology for direct communications among devices furthermore provides one more mechanism for network offloading and is a new useful tool for social networking. D2D is expected to be a key technology to improve system capacity and user experience in various service scenarios as LTE-D2D is being positioned for emergency services. Although D2D is now on the way towards standardization through 3GPP but still under development and in spite of the numerous benefits offered by D2D communication, there are many technical issues including how to coexist with cellular network users and how to deal with interferences are still being unresolved and thus a fertile ground for research. When sharing the same resources, interference between the cellular users and D2D users needs to be controlled. A number of concerns are involved with its implementation whereas we need to develop D2D applications which are attractive to both operators and users. Peer discovery and mode selection, power control for the devices, radio resource allocation and security of the communication are the other concerns that should be mentioned. These are open issues which proposed potential future research directions.

REFERENCES

[1] J. Liu, N. Kato, J. Ma, and N. Kadowaki, Device-to-Device Communication in LTE-Advanced Networks: A Survey, IEEE Commun. Surv. Tutorials, vol. 17, no. 4, pp. 19231940, 2015.

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20162021 White Paper - Cisco. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. [Accessed: 06-Jan-2019].

[3] A. Asadi, Q. Wang, and V. Mancuso, A survey on device-to-device communication in cellular networks, IEEE Commun. Surv. Tutorials, vol. 16, no. 4, pp. 18011819, 2014.

[4] B. Jedari, F. Xia, and Z. Ning, A Survey on Human-Centric Communications in Non-Cooperative Wireless Relay Networks, IEEE Commun. Surv. Tutorials, vol. 20, no. 2, pp. 914944, 2018.

[5] A. Gupta and R. K. Jha, A Survey of 5G Network: Architecture and Emerging Technologies, IEEE Access, vol. 3. pp. 12061232, 2015.

[6] G. Fodor et al., Design aspects of network assisted device-to-device communications, IEEE Commun. Mag., vol. 50, no. 3, pp. 170177, Mar. 2012.

[7] Lei Lei, Zhangdui Zhong, Chuang Lin, and Xuemin Shen, Operator controlled device-to-device communications in LTE-advanced networks, IEEE Wirel. Commun., vol. 19, no. 3, pp. 96104, Jun. 2012.

[8] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, G. Feng, and S. Li, Device-to-Device Communications Underlaying Cellular Networks, IEEE Trans. Commun., vol. 61, no. 8, pp. 35413551, Aug. 2013.

[9] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, and K. Hugl, Device-to-device communication as an underlay to LTE-advanced networks, IEEE Commun. Mag., vol. 47, no. 12, pp. 4249, Dec. 2009.

[10] Wei Xu, Le Liang, Hua Zhang, Shi Jin, J. C. F. Li, and Ming Lei, Performance enhanced transmission in device-to-device communications: Beamforming or interference cancellation?, in 2012 IEEE Global Communications Conference (GLOBECOM), 2012, pp. 42964301.

[11] Rongqing Zhang, Xiang Cheng, Liuqing Yang, and Bingli Jiao, Interference-aware graph based resource sharing for device-to-device communications underlaying cellular networks, in 2013 IEEE Wireless Communications and Networking Conference (WCNC), 2013, pp. 140145.

[12] Qing Wang and B. Rengarajan, Recouping opportunistic gain in dense base station layouts through energy-aware user cooperation, in 2013 IEEE 14th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013, pp. 19.

[13] Chia-Hao Yu, K. Doppler, C. B. Ribeiro, and O. Tirkkonen, Resource Sharing Optimization for Device-to-Device Communication Underlaying Cellular Networks, IEEE Trans. Wirel. Commun., vol. 10, no. 8, pp. 27522763, Aug. 2011.

[14] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects;, Proximity-based services (ProSe); Stage 2, no. Release 12, 2014.

[15] G. Fodor, S. Parkvall, S. Sorrentino, P. Wallentin, Q. Lu, and N. Brahmi, Device-to-Device Communications for National Security and Public Safety, IEEE Access, vol. 2, pp. 15101520, 2014.

[16] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects;, Study on architecture enhancements to support Proximity-based Services (ProSe), no. Release 12, pp. 117, 2014.

[17] IEEE Computer Society. LAN/MAN Standards Committee., Institute of Electrical and Electronics Engineers., and IEEE-SA Standards Board., IEEE standard for Information technology– telecommunications and information exchange between systems– local and metropolitan area networks– specific requirements: Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specificati. Institute of Electrical and Electronics Engineers, 2010.

[18] F. Ghavimi and H.-H. Chen, M2M Communications in 3GPP LTE/LTE-A Networks: Architectures, Service Requirements, Challenges, and Applications, IEEE Commun. Surv. Tutorials, vol. 17, no. 2, pp. 525549, 2015.

[19] K. Ali, H. X. Nguyen, P. Shah, Q. T. Vien, and N. Bhuvanasundaram, Architecture for public safety network using D2D communication, in 2016 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2016, 2016, pp. 206211.

[20] M. Hunukumbure, T. Moulsley, A. Oyawoye, S. Vadgama, and M. Wilson, D2D for energy efficient communications in disaster and emergency situations, in 2013 21st International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2013, 2013, pp. 15.

[21] K. Ali, H. X. Nguyen, Q. T. Vien, P. Shah, and Z. Chu, Disaster Management Using D2D Communication with Power Transfer and Clustering Techniques, IEEE Access, vol. 6, pp. 1464314654, 2018.

[22] A. Alnoman and A. Anpalagan, On D2D communications for public safety applications, in 2017 IEEE Canada International Humanitarian Technology Conference (IHTC), 2017, pp. 124127.
[23] J. Lianghai, B. Han, M. Liu, and H. D. Schotten, Applying Device-to-Device Communication to Enhance IoT Services, IEEE Commun. Stand. Mag., vol. 1, no. 2, pp. 8591, 2017.
[24] L. Militano, G. Araniti, M. Condoluci, I. Farris, and A. Iera, Device-to-Device Communications for 5G Internet of Things, EAI Endorsed Trans. Internet Things, vol. 1, no. 1, p. 150598, Oct. 2015.
[25] Y. Zhang, E. Pan, L. Song, W. Saad, Z. Dawy, and Z. Han, Social network aware device-to-device communication in wireless networks, IEEE Trans. Wirel. Commun., vol. 14, no. 1, pp. 177190, Jan. 2015.
[26] X. Lin, J. Andrews, A. Ghosh, and R. Ratasuk, An overview of 3GPP device-to-device proximity services, IEEE Commun. Mag., vol. 52, no. 4, pp. 4048, Apr. 2014.
[27] M. Natkaniec, Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications (Sarkar, S. K. et al.; 2008) [Book Review], IEEE Commun. Mag., vol. 47, no. 5, pp. 1214, May 2009.
[28] K. Ali, H. X. Nguyen, P. Shah, Q.-T. Vien, and N. Bhuvanasundaram, Architecture for public safety network using D2D communication, in 2016 IEEE Wireless Communications and Networking Conference, 2016, pp. 16.

Comparative Analysis of Probability of Error for Selected Digital Modulation Techniques

Ehsan Shahri
FEUP, University of Porto
Porto, Portugal
ehsan.shahri@fe.up.pt

***Abstract*—To select a suitable modulation in digital communication systems, various parameters such as Bit Error Rate (BER), Signal to Noise Ratio (SNR), available bandwidth, power spectral density and etc. are considered. Therefore, the performance of modulation is based on the probability of the error parameter. If modulation is capable of sending more data and having the lowest noise in the output, it will have high performance. In this paper, some of the modulators are simulated such as PSK, DPSK, FSK, MSK, MPSK, and MQAM using MATLAB, and their BER performances are calculated. Cyclic and convolutional error correction code are also implemented on a multipath channel with a long excess delay and their BER performances are compared with previous modulators. Finally, a form of OFDM modulation is designed on a direct path with three obstacles and parameters of a multipath channel are adapted on this path. The simulation results are illustrated that BPSK modulation has higher performance than other binary modulations. In M-ary transmission, MQAM has a better performance than MPSK as well as the performance of convolutional coding is preferable over cyclic coding. However, the results confirmed that the proposed OFDM method improves the BER performance in the multipath channel.**

***Index Terms*—BER, binary modulations, M-ary transmission, cyclic code, convolutional code, OFDM**

I. Introduction

A digital communication system is a system where the information signal sent from A to B can be fully described as a digital signal. In this system, data is modulated and then is transferred to channel. Without modulation, all signals at the same frequencies from different transmitters would be mixed up. In order to separate the various signals, stations must broadcast the data at different frequencies. Each station must be given its own frequency band. This is achieved by frequency translation as a result of the modulation process. Modulation is an important part of the communication system. Modulation is defined as the process whereby some characteristic (amplitude, frequency, phase) of a high-frequency signal wave is varied in accordance with the instantaneous value intensity of the low-frequency signal wave. Hence, there are three basic types of modulation: Amplitude modulation, Frequency modulation, and Phase modulation. Nowadays, a lot of information is sent digitally, and the election of modulation type is important due to the spectrum constraints. In digital transmitting of data, some parameters such as the system reliability, performance of the transition, available bandwidth, information security, system capacity and the probability of error are reviewed [1], [2]. Digital modulations are capable of sending a large quantity of data with high capacity and noise immunity. They also have the ability to detect and correct errors in receivers [3]. In digital transmission, the number of bit errors is the number of received bits of a data stream over a communication channel that has been altered due to noise, interference, distortion or bit synchronization errors. The Bit Error Rate (BER) is the number of bit errors per unit time [4]. In other words, the performance of communication systems is measured through the value of the BER. The most important approach in digital communication systems is the maximum use of the bandwidth and the lowest probability of error. Modulation should be able to use a limited bandwidth according to the amount of power efficiency [5]. There are many barriers to sending data in real-world environments such as cities, highways, and villages to send data directly, and problems like fading signals, intersymbol interference are created. Therefore, in this paper, different modulations are implemented and probabilities of error are estimated to select the best modulators for implementation in real environments. The proposed OFDM modulation provides the best performance for multipath channel transmission. The rest of this paper is organized as follow. Section II presents the modulation techniques in digital communication systems. Implementation methods of modulation techniques are discussed in section III. Simulation results and evaluation of BER performance are presented in section IV. The final conclusion of this paper is discussed in section V.

II. Modulation Techniques in Digital Communication Systems

Digital modulation [6] is used to transfer a digital bit stream over an analog channel at a high frequency. This enables us to transmit signals generated in a digital circuit across a physical medium. In digital modulation, an analog carrier signal is modulated by a discrete signal. Digital modulation schemes are as a form of digital transmission, synonymous to data transmission. Any digital modulation scheme uses a finite number of distinct signals to represent digital data. There are most common digital modulation techniques, including Phase-shift keying (PSK) [6], Frequency-shift keying (FSK) [7], Amplitude-shift keying (ASK) [7], On-off keying (OOK) [8] which the most common ASK form, Quadrature amplitude modulation (QAM) [9] which a combination of PSK and ASK, Continuous phase modulation (CPM) methods [10], Orthogo-

nal frequency-division multiplexing modulation (OFDM) [11], [12], Wavelet modulation [13], Trellis coded modulation (TCM) [14] and finally Spread spectrum techniques [15]. Each of these methods also has a number of techniques that have different characteristics. The modulation techniques that are simulated in this paper are as follow:

A. Phase-shift keying (PSK)

Phase-shift keying (PSK) is a method for modulating a digital signal which transmits the data by modulating the carrier phase. The phase of the carrier signal is diverse to represent binary 1 or 0. This method sends binary data digitally using a limited number of digits allocated to the digital phase. In other words, each character encodes a number of data by changing or modulating their phases. Amplitude and frequency are remained constant during each bit interval. For demodulation, the phase of the received signal is determined and the main data is extracted. Therefore, the system must be able to compare the received signal phase with the original signal. In PSK, the output phase signal is shifted according to the input signal. Hence, this method is classified according to the number of phase shifts to divisions Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK). Generally, the basic binary transmission uses one bit per symbol. BPSK uses two phases which are separated by 180° and hence the constellation diagram of BPSK will show the constellation points on the real axis at 0° and 180°. Therefore, due to this property, it handles the highest noise level or distortion before the demodulator reaches an incorrect decision. QPSK is a form of phase modulation technique where two data bits that are located on a symbol, are modulated at the same time and it uses one of the four available phase shifts. In M-ary transmission two or more bits are transmitted at the same time instead of transmitting one bit at a time and hence, channel bandwidth is reduced. In this kind of modulators, signals are generated by deferent changing the amplitude, phase or frequency of a carrier in M discrete signal. In many cases, the combination of these changing methods is also used for generating signals. In M-ary Phase Shift Keying (MPSK), data bits select one of M phase shifted versions of the carrier to transmit the data which in this case there are M possible waveforms that have the same amplitude and frequency with different phases. Also, in M-ary Quadrature Amplitude Modulation (MQAM), data bits select one of M combinations of phase shifts and amplitude that are applied to the carrier signal. The constellation diagram of BPSK and QPSK which represent the modulated signal by a digital modulation scheme is shown in Fig. 1. In BPSK constellation, two phases are on the real x-axis, and in the QPSK constellation, the phases are divided into four regions of 90 degrees.

The general form of transmitted signal for BPSK and QPSK follows the (1) and (2) respectively where the Eb is energy per bit, Es is energy per symbol with n bits and T is time duration.

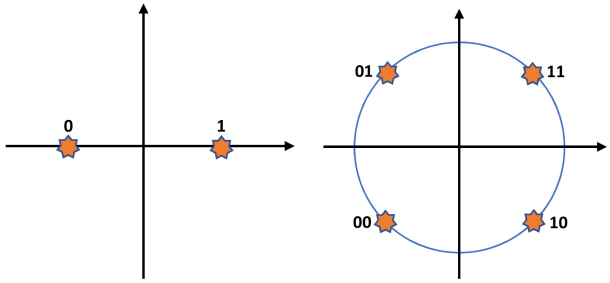


Fig. 1. The constellation diagram of BPSK (Left diagram) and QPSK (Right diagram).

$$\begin{cases} S_1(t) = \sqrt{\frac{2E_b}{T}}\cos(\omega_c t + \phi) & (\phi = 0) \\ & (0 \le t \le T) \\ S_2(t) = \sqrt{\frac{2E_b}{T}}\cos(\omega_c t + \phi + \pi) = -S_1 \end{cases} \quad (1)$$

$$S_i(t) = \sqrt{\frac{2E}{T}}\cos(2\pi f_c t - \frac{(2i-1)\pi}{4}) \quad \begin{cases} f_c = n_c\frac{1}{T} \\ i = 1,2,3,4, \\ (0 \le t \le T) \end{cases} \quad (2)$$

The probability of bit error for QPSK is the same as for BPSK but due to QPSK uses two bits to transmit, therefore, QPSK uses twice the power. The general equations of the probability of error for BPSK and QPSK are defined by (3) and (4) respectively where No is noise power spectral density (W/Hz).

$$P_b = Q(\sqrt{\frac{2E_b}{N_o}}) \quad (3)$$

$$P_b(QPSK) = \frac{P_e}{2} \approx Q(\sqrt{\frac{2E_b}{N_o}}) = P_b(BPSK) \quad (4)$$

BPSK and QPSK power spectrum are defined by (5) and (6) respectively where the Tb is bit duration and Es is energy per symbol with n bits.

$$S_B(f) = 2E_b sinc^2(T_b f) \quad (5)$$

$$S_B(f) = 2E_s sinc^2(Tf) = 4E_b sinc^2(2T_b f) \quad (6)$$

Although more efficient use of bandwidth (higher data-rate) are possible in BPSK it has more complex signal detection and recovery process against other modulations.

B. Differential PSK (DPSK)

Differential PSK (DPSK) [6] is a common method with PSK which the data is sent by modulating the phase. In this method, the data is used to modulate instead of the phase and thus the ambiguity problem is solved in the phase rotation (Due to the constellation rotation). In fact, in DPSK the phase is changed according to the previous value. In better words, in DPSK a binary "1" number is sent by adding 180° degrees to the phase and a binary "0" number is sent by adding 0° degrees to the

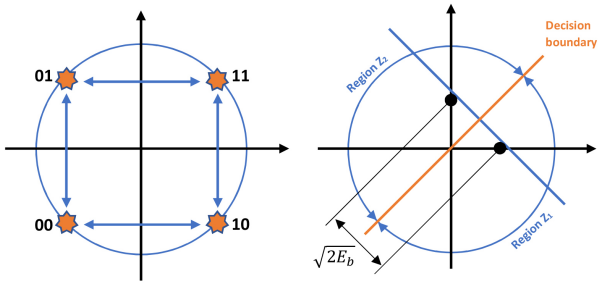


Fig. 2. The constellation diagram of MSK(Left diagram) and BPSK(Right diagram) modulations.

current phase. For demodulation in differential PSK, the two received symbols in the output are compared with each other and the corrected signal is identified. As mentioned, DPSK is common to BPSK modulation, therefore, its constellation is similar to the constellation diagram of BPSK modulation. The general form for DPSK follows the (7).

$$\begin{cases} S_1(t) = \sqrt{\frac{2E_b}{T_b}}\cos(\omega_c t) & (0 \le t \le T_b), (T_b \le t \le 2T_b) \\ S_2(t) = \sqrt{\frac{2E_b}{T_b}}\cos(\omega_c t)) & (0 \le t \le T_b) \\ S_2(t) = \sqrt{\frac{2E_b}{T}}\cos(\omega_c t + \pi) & (T_b \le t \le 2T_b) \end{cases} \quad (7)$$

The general equation of the probability of error for DPSK is defined by (8). Although the error rate has almost doubled using DPSK, this can be defeat by increasing a little Eb/No.

$$P_b = \frac{1}{2}e^{-\frac{E_b}{N_o}} \quad (8)$$

C. Minimum-shift keying (MSK)

Minimum-shift keying (MSK) [16] which is a type of continuous-phase frequency-shift keying, encodes signals with bits alternating between quadrature components with the Q component delayed by half the symbol period. MSK reduces the distortion that it caused by nonlinear systems because it codes each bit as half the sinusoid. In this case, it can show the signal with continuous frequency. It should be noted that the difference between the highest and lowest frequencies in this method, is the same as the waveform which used to display 0 and 1 bits differ by exactly half a carrier period. The constellation diagram of MSK is shown in Fig. 2. The general form for MSK modulation follows the (9).

$$S(t) = \sqrt{\frac{2E_b}{T_b}}\cos\theta(t)\cos(2\pi f_c t) - \sqrt{\frac{2E_b}{T_b}}\sin\theta(t)\sin(2\pi f_c t) \quad (9)$$

The equations of probability of error and power spectrum for MSK are defined by (10) and (11) respectively.

$$P_b = Q\sqrt{\frac{2E_b}{N-o}} \quad (10)$$

$$S_B(f) = \frac{32E_b}{\pi^2}[\frac{\cos(2\pi T_b f)}{16T_b^2 f^2 - 1}]^2 \quad (11)$$

D. Frequency-shift keying (FSK)

In Frequency-shift keying (FSK), data is sent by modulating the frequencies of the signal. In FSK a finite number of frequencies are used while amplitude and phase remain constant during each bit interval. In this method, two discrete frequencies are used to send data signals that the frequency of the carrier signal is divided to represent binary 1 or 0. The binary "1" is the mark frequency and the "0" is the space frequency. The frequency of the carrier signal is divided to represent binary 1 or 0. Goertzel algorithm [17] is used to modulate binary signals in FSK modulation which greatly increases the efficiency of the system. The demodulator of FSK must be able to determine two possible frequencies which are presented at a given time. The constellation diagram of BFSK is shown in Fig. 2. The general form of transmitted signal for binary frequency-shift keying (BFSK) with continuous phase is defined as (12).

$$\begin{cases} S_i(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_i t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t \pm \frac{\pi}{T_b}t) \\ i = 1, 2 \qquad (0 \le t \le T_b) \end{cases} \quad (12)$$

The equation of probability of error and power spectrum for MFSK is defined by (13) and (14) respectively. Although, the receiver is looking for frequency variations in FSK modulation, it has a slight sensitivity to error. For this reason, it abandons the edges of the noise spikes.

$$P_b = Q(\sqrt{\frac{2E_b}{N_o}}) \quad (13)$$

$$S_B(f) = \frac{E_b}{2T_b}[\delta(f - \frac{1}{2T_b}) + \delta(f + \frac{1}{2T_b})] + \frac{8E_b\cos^2\pi T_b f}{\pi^2(4T_b^2 f^2 - 1)^2} \quad (14)$$

E. Quadrature amplitude modulation (QAM)

Quadrature amplitude modulation (QAM) is also another digital modular technique. In this method, data is sent digitally using ASK by modulating the amplitude of the two carrier signals. Two carrier signals that have the same frequency send data at an orthogonality and quadrature condition as well as in the fixed phase. Due to of their orthogonality property the modulated signals can be sent at the same frequency and can be coherently demodulated at the output. QAM modulations are low-frequency and low-bandwidth waveforms against the carrier frequency signal. Input information stream is divided into two sequences that consist of odd and even symbols. The first symbol is modulated by sin term and the second symbol is modulated by cos term. Sum of first and second sequences are sent to the channel. In this situation, the data rate is two bits per bit-interval. By dividing the output signal by the first and second symbols sequences and passing them through the low pass filters, the main signal is obtained. The constellation diagram of QAM is shown in Fig. 3. The general form of the transmitted signal for QAM modulation follows the (15). The equations of the probability of error for QAM is defined by (16).

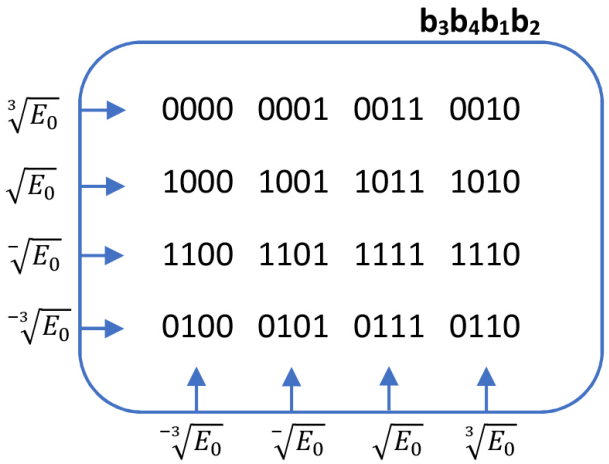


Fig. 3. The constellation diagram of QAM modulation.

$$\begin{cases} S_i(t) = \sqrt{\frac{2E_b}{T_b}} a_i \cos(2\pi f_c t) + \sqrt{\frac{2E_b}{T_b}} a_i \sin(2\pi f_c t) \\ i = -L+1, ..., -1, 0, 1, ..., L-1, \quad (0 \le t \le T) \end{cases} \tag{15}$$

$$P_b \approx \frac{4(1-\frac{1}{\sqrt{M}})}{\log_2 M} Q(\sqrt{\frac{3\log_2 M}{M-1}\frac{E_b}{N_o}}) \tag{16}$$

F. Orthogonal frequency-division multiplexing modulation (OFDM)

Orthogonal frequency-division multiplexing modulation (OFDM) is another technique for encoding digital data on multiple carrier frequencies. OFDM is a technique for transmitting large amounts of digital data over a radio wave. This technology works by splitting the radio signal into multiple smaller sub-signals that are then transmitted simultaneously at different frequencies to the receiver. OFDM provides better orthogonality in transmission channels affected by multipath propagation through using guard interval. OFDM modulates without the use of complex equalization filters because this modulation may be used as a slow modulator in a narrowband instead of a fast modulator in wideband. Hence, this technic is able to overcome channels with hard conditions such as narrowband interference, fading signals and etc. The low symbol rate uses a guard interval between symbols affordable and therefore, it makes possible to eliminate intersymbol interference. OFDM demodulation is based on Fast Fourier Transform algorithms. The constellation diagram of OFDM is shown in Fig. 4.

III. IMPLEMENTATION METHODS OF MODULATION TECHNIQUES

The modulation techniques are designed using MATLAB to allow implementation of various parameters and situation of the system. The simulations process generally use the same simple scenario for all digital modulators. First, random data is generated, and then the generated data is modulated according to the modulation type and their properties. Subsequently, the modulated data is sent to the channel and finally, the output

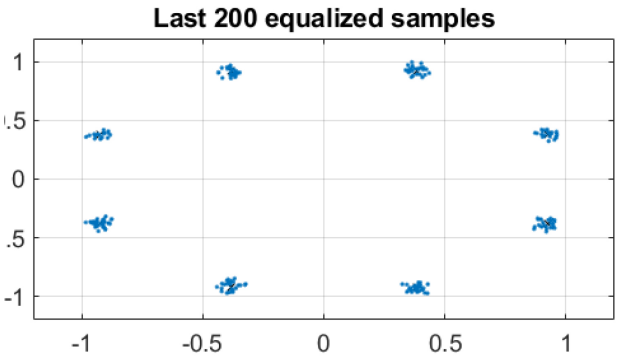


Fig. 4. The constellation diagram of OFDM modulation.

TABLE I
THE COMMON PARAMETERS IN SIMULATIONS TECHNICS

General parameters	Specifications
Number of symbols	2^20
Eb/Np Ratio	0 up to 15 (dB)
M-ary values	M-PSK = [4 8 16 32]
M-ary values	M-QAM = [4 16 64]
Cyclic code parameters	(Desired signal) k = $\log_2$(MPSK)-1 (Codeword bits) n = $\log_2$(MPSK)
Convolutional code parameters	Code Rate = 2/3 Consternate = 3 Code Generator = [7 5]
Variable attenuations for multipath channel	att1=0.3 att2=0.2 att3=0.45 att4=0.5
OFDM modulation parameters	FFT size= 64 Number of subcarriers= 52 Number of bits= 52

data is demodulated and BER is calculated. The common parameters that are used in all of these simulations are specified in Table I.

A. Binary digital transmition

To use and simulate modulators and demodulators, random data must first be generated. After generating random data and defining the number of symbols per bit, they are modulated by some of the binary modulators as mentioned in the previous section. The generated signal has been transmitted through an AWGN channel in order to add White Gaussian Noise to the signal because when the data is sent in a communication channel, different intrinsic noise is added to the signal. After transmitting the data in an AWGN channel, the modulated signal that contains the noise in the output is demodulated and the number of symbol errors is calculated. By counting the number of symbol errors, the BER is calculated in different values of the signal-to-noise ratio (SNRs). In finally, probabilities of error for various modulations are calculated.

B. M-ary transmition

In M-ary modulation, the values of M in MPSK and MQAM modulations are considered according to the Table I. To

compare the BER performance of the modulations MPSK and MQAM, the SNR is defined by the baseband bit energy (Eb) over the noise power spectral density No which is shown in (17). In this modulation, the L value bits per symbol is transmitted to the channel where M is the constellation number of symbols.

$$SNR = (\frac{E_b}{N_o}) + 10\log_{10}(L), \quad L = (\log_2 M) \tag{17}$$

C. The cyclic error correction code

A cyclic code is a block code where the circular shifts of each codeword give another word that belongs to the code. They are error-correcting codes that use algebraic properties to efficiently detect and correct errors. In this technique, redundancy is added to the original bit sequence to increase the reliability of the communication. This simulation is the same as the previous simulation but in this situation, a cyclic code is used. After defining cyclic code parameters, SNR and random data symbols, the input decimal data is converted to binary data. Then the input data is encoded by cyclic code and afterward modulated. Although modulation and demodulation are easier with the use of the cyclic code, data must be coded using cyclic techniques before sending data to the channel. The encode function receives integer values between 0 and M-1 rather than L. Afterward the encoded signals are modulated and sent to AWGN channel. Finally, the modulated code is demodulated and the output is converted to integer-valued data symbols form. After generating the output binary signals, the bit errors are detected and corrected.

D. The convolutional error correction codes

A convolutional code is a type of error-correcting code that generates parity symbols via the sliding application of a Boolean polynomial function to a data stream. The sliding application represents the 'convolution' of the encoder over the data which gives rise to the term 'convolutional coding'. This parity symbol is used by the decoder to infer the message sequence by performing error correction. Convolutional codes are defined based on three parameters including base code rate, constraint length, and generator polynomial. The base code rate is typically given as n/k where n is the input data rate and k is the output symbol rate. The depth is often called the "constraint length" K and finally, the output connections for each of the encoder input is related to generator polynomial. In this simulation, the input data is encoded according to Table I parameters. By the convolutional method, the data is first coded. Afterward, the input signal is reshaped into a binary matrix and then binary vectors are converted to decimal numbers. The code is converted into its trellis representation and then modulated. Finally, data is transmitted to a white Gaussian noise channel. The Viterbi algorithm is used to decode the received data. Since the decoding operation in decoder has a delay time, the Viterbi decoder also has a delay time that is equal to the traceback length. Therefore the traceBack for code rate 2/3 is considered equal to 7.5(k – 1), where k is the constant length.

E. The cyclic and convolutional error correction code in multipath channel

In multipath propagation channels, the signals are reached to the receiving antenna by two or more paths and signals on these paths encounter various obstacles. Multipath propagation causes multipath interference (ISI) including constructive and destructive interference. This phenomenon produces a noise that it makes communication with less reliability. When a signal in the multipath channel reaches by different paths with different lengths to the receiver, they have different delay times. In addition, the various paths often distort the amplitude and/or phase of the signal and therefore, this effect will cause more interference in the received signal. When the phases of arriving signals have changed, may lead to significant changes in the total received power. To simulate data by cyclic and convolutional error correction code in a multipath channel, first, a direct path with three reflections paths are designed according to parameters of Table I. Input decimal data is converted to binary data and then encoded by cyclic or/and convolutional code. Afterward, the encoded data is modulated and then sent to the AWGN channel. The received signal in the receiver is equalized and then enters the demodulator for demodulation. Finally, the data bits are decoded and BER is calculated. The received signal will be given by (18).

$$r(t) = \sum a_k(t)S(t - T_k) \tag{18}$$

F. OFDM modulation in multipath channel

In OFDM, N subcarriers are intended that each subcarrier is centered at frequencies which are orthogonal to each other. Based on these N subcarriers, the data is converted from serial to a parallel stream. The serial to parallel converter takes the serial stream of input bits and produces N parallel streams. Then, these parallel streams are modulated individually based on the modulation format such as BPSK, QPSK, QAM. Hence, the modulated symbols are assigned to the required orthogonal subcarriers. The number of total presented subcarriers in the OFDM system is specified by the FFT/IFFT length N. Not only all of these sub-carriers aren't used to transmit data, but some of them are reserved for pilot carriers and some of them are used to act as a guard band. It should be noted that in order to overcome the problems which are caused by the multipath propagation such as Inter-Symbol Interference (ISI), a cyclic prefix should be added to each OFDM symbol. Finally, according to the given points and also the following parameters, the input signal is sent using OFDM modulation on the multipath channel and then demodulated in the receiver. Specifications of designed OFDM transmitter are specified in Table II.

IV. SIMULATION RESULTS AND EVALUATIONS

In this section, the simulation results of binary digital modulations, MPSK and MQAM, cyclic and convolutional error correction as well as cyclic code, convolutional code and OFDM modulation in the multipath channel are presented.

TABLE II
SPECIFICATIONS OF OFDM TRANSMITTER MODEL DESIGN

Key Parameters of the OFDM transmitter	Specifications
Number of used sub-carriers	52
OFDM symbol duration	4 µsec
Guard interval (cyclic prefix)	0.8 µsec
Subcarrier Spacing	312.5 kHz
Channel bandwidth	20 MHz

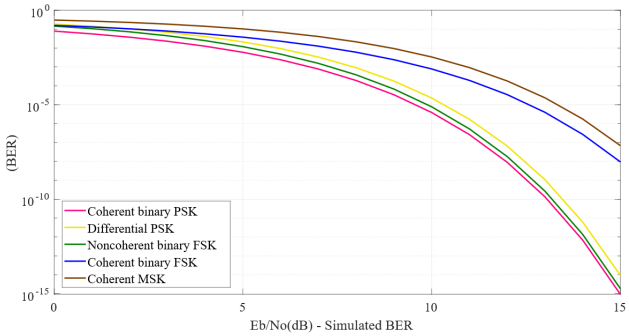


Fig. 5. Probabilities of error for various binary digital modulations.

A. Estimation of the probabilities of error for the various binary digital modulations

The input data is modulated using binary modulation techniques including coherent binary PSK, differential PSK, coherent binary MSK, coherent binary FSK and non-coherent BFSK. The probabilities of error for various binary modulations are shown in Fig. 5. Coherent binary PSK has the best performance of the BER compared to other modulators because in this modulator the symbols are more distant from each other, and thus present a better Pe for the same Eb/No. Coherent MSK is a little worse than coherent binary PSK. According to the graph, coherent binary FSK is a little better than noncoherent FSK and finally, differential PSK is better than coherent binary FSK. When the noncoherent binary FSK is worse than of its coherent, this means that noncoherent -FSK needs more transmission power than its coherent to achieve the same power spectral density.

B. Estimation of the probabilities of error for the modulations MPSK and MQAM

In this part, the M-PSK and M-QAM modulations are simulated and the BERs are calculated. The probabilities of error for MPSK and MQAM are shown in Fig. 6. The graph illustrates by increasing M, the number of phases, the performance of the symbol error rate and the performance of error are increased, but the value of the bandwidth is decreased. According to Fig. 6, BPSK and QPSK require the same rate of (Eb/No) to achieve the same probabilities of error. This is while QPSK has a better bandwidth efficiency. It illustrates the MQAM works better than MPSK because in MQAM the distance between two signal points neighbors is higher than in MPSK.

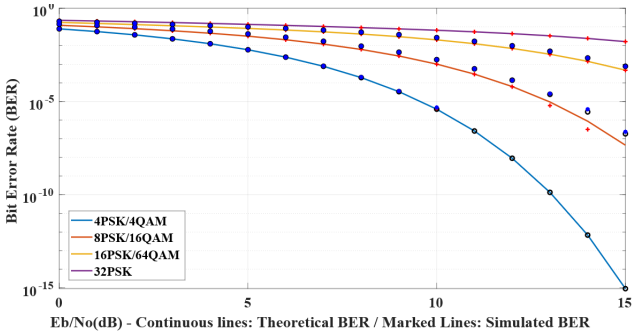


Fig. 6. Probabilities of error for MPSK and MQAM modulations.

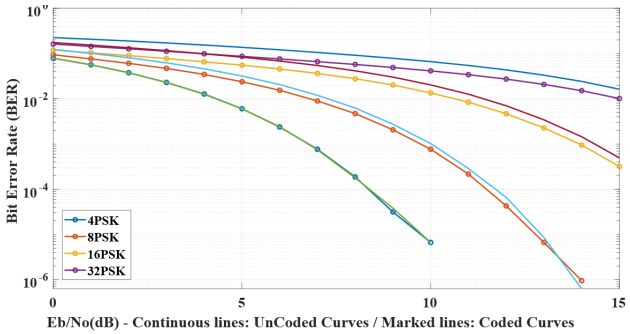


Fig. 7. Probabilities of error for MPSK using and without cyclic error correction code.

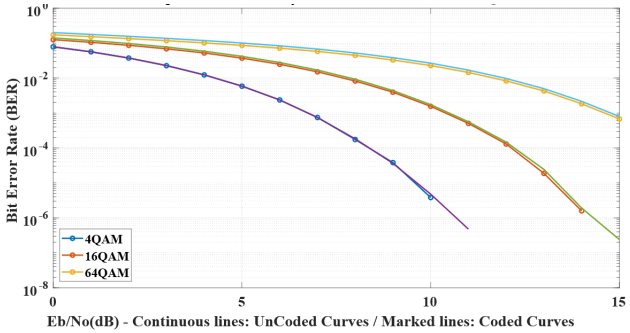


Fig. 8. Probabilities of error for MQAM using and without cyclic error correction code.

C. Estimation of the probabilities of error for the cyclic error correction code

This part is the same as the previous simulation but in this situation, a cyclic code is used. The obtained results for the MPSK and MQAM modulations technics using cyclic error correction code and without cyclic error correction code are depicted in Fig. 7 and Fig. 8 respectively. By comparing the results of this method with previous methods, it has been determined that the BER performance of the system has increased by using the Cyclic method, especially with increasing the number of M.

D. Estimation of the probabilities of error for the convolutional error correction codes

In this part, the convolutional error correction code is used to simulate the input data. Probabilities of error for various MPSK and MQAM using and without convolutional error

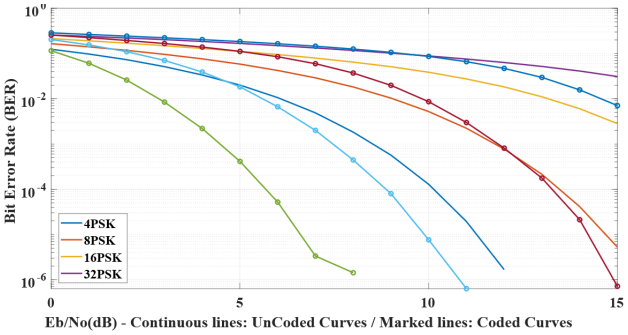


Fig. 9. Probabilities of error for MPSK using and without convolutional error correction code.

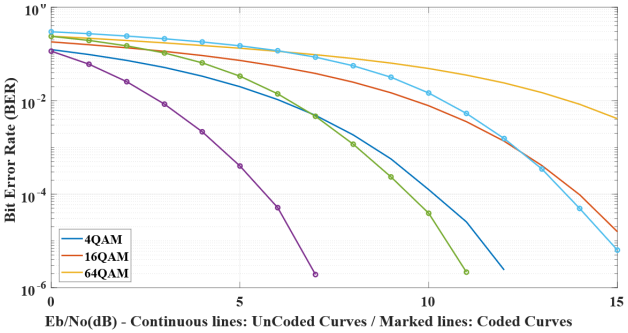


Fig. 10. Probabilities of error for MQAM using and without convolutional error correction code.

correction code are shown in Fig. 9 and Fig. 10 respectably. These graphs illustrate that for fewer quantities of (Eb/No), the BER without coding is less than of the BER with Viterbi decoding because for fewer quantities of (Eb/No), there are more chances of multiple received coded bits in error and the (Viterbi) algorithm is unable to recover the signal. Compared to the cyclic code, the performance of the system has increased by using the convolutional method.

E. Estimation of the probabilities of error for cyclic error correction for MPSK and MQAM in multipath channel

In this part, the properties of the proposed multipath channel are affected by the previous simulation. Hence, this process is repeated for cyclic and convolutional error correction code by the previous experiment materials. The probability of error for the modulated signal with MPSK and MQAM which are transmitted through a multipath fading channel, are shown in Fig. 11 and Fig. 12 respectively. Although the cyclic error correction can be able to increase the performance of the system for MPSK and MQAM, these graphs show the BER performance in the multipath channel is decreased.

F. Estimation of probabilities of error for convolutional error correction for MPSK and MQAM in multipath channel

In this part, probabilities of error for various MPSK and MQAM in a multipath channel, with and without convolutional error correction code are shown in Fig. 13 and Fig. 14, respectively. As previously noted, the multipath channel has different delay times and also produces a noisy effect on the signal. This error effect is well illustrated in the obtained graphs. The inappropriate performance of MQAM and MPSK is clearly observable in simulation graphs.

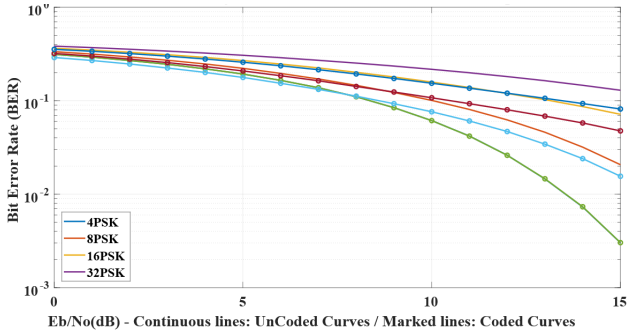


Fig. 11. Probabilities of error for MPSK in a multipath channel, with and without cyclic error correction.

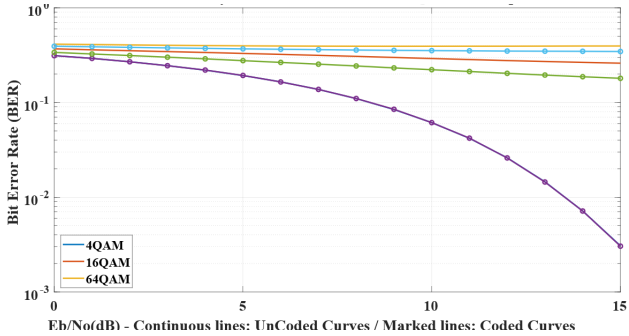


Fig. 12. Probabilities of error for MQAM in a multipath channel, with and without cyclic error correction.

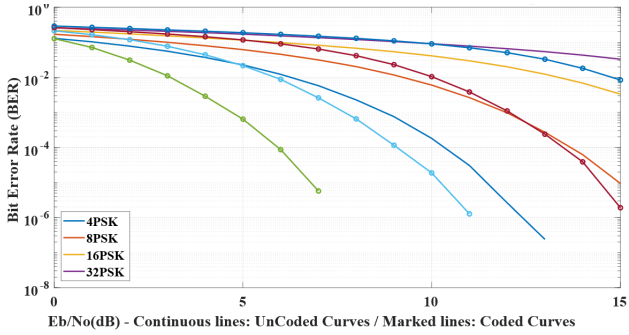


Fig. 13. Probabilities of error for MPSK in a multipath channel, with and without convolutional error correction code.

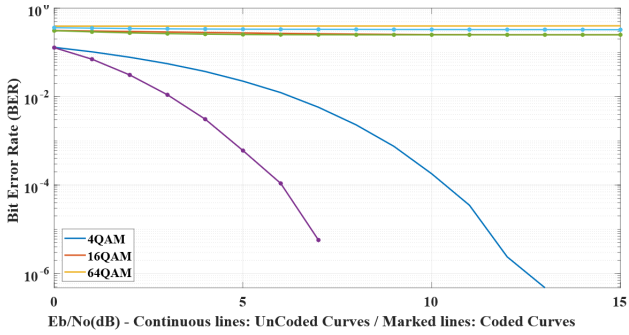


Fig. 14. Probabilities of error for MQAM in a multipath channel, with and without convolutional error correction code.

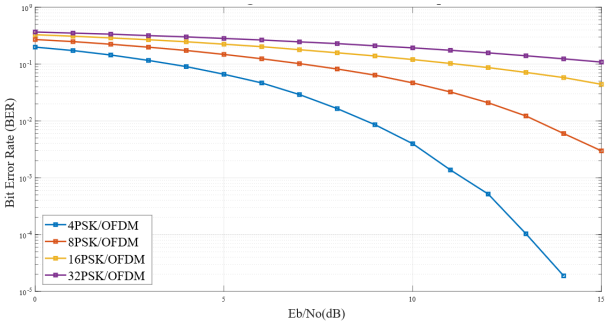


Fig. 15. Probabilities of error for MPSK using OFDM modulation in a multipath channel.

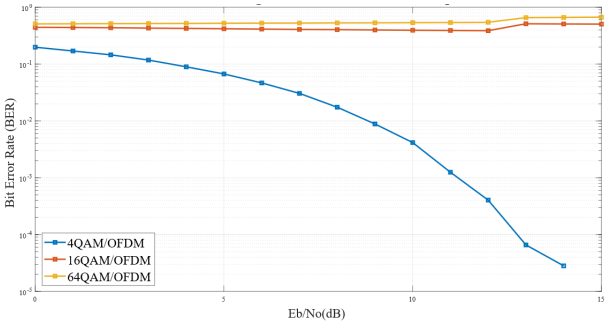


Fig. 16. Probabilities of error for MQAM using OFDM modulation in a multipath channel.

G. Estimation of the probabilities of error for the MPSK and MQAM using OFDM modulation in a multipath channel

In this part, the simulation is developed for computing the BER for MPSK in OFDM modulation over a multipath channel. Probabilities of error for MPSK and MQAM using OFDM modulation in a multipath channel are shown in Fig. 15 and Fig. 16 respectively. The simulation graphs show the performance of BER for MPSK and MQAM using OFDM modulation is increased in a multipath channel.

V. CONCLUSIONS

BER performance is one of the important parameters to select a suitable modulation in digital communication systems. If modulation is capable of sending more data and having the lowest noise in the output, it will have high performance. Many digital transmissions in real environments are based on multiple channels. Multipath propagation causes multipath intersymbol interference (ISI) including constructive and destructive interference. This phenomenon produces a noise that it makes communication with less reliability. In this paper, first some of the modulators are simulated and then the superiority and benefits of OFDM modulation in the proposed multipath channel are discussed. In this simulation scenario, it became clear that when the binary modulation is used, BPSK modulation will have higher performance. Also, for M-ary data transmission, M-QAM modulation has a better performance than M-PSK modulation. The convolutional coding is preferable over cyclic coding because it allows higher error correction and this is while the BER performance is improved by using forward error correction coding. Although the BER for MQAM and MPSK over multipath channels are increased, the performance of BER using OFDM modulation is increased in a multipath channel. Hence, OFDM modulation can be effective in applications such as digital television and audio broadcasting, DSL internet access, wireless networks, power line networks, and 4G mobile communications. To increase system performance as future work, investigating the influence of other parameters in the simulation environment and, in the next step, the actual system is suggested.

REFERENCES

[1] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996, vol. 2.

[2] M. Barnela and D. S. Kumar, "Digital modulation schemes employed in wireless communication: A literature review," *International Journal of Wired and Wireless Communications*, vol. 2, no. 2, pp. 15–21, 2014.

[3] R. Pandey and K. Pandey, "An introduction of analog and digital modulation techniques in communication system," *Journal of Innovative Trends in Science Pharmacy & Technology*, vol. 1, 2014.

[4] M. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems," *IEEE Journal on selected areas in communications*, vol. 2, no. 1, pp. 153–170, 1984.

[5] L. E. Kopp, "Method and system to increase the throughput of a communications system that uses an electrical power distribution system as a communications pathway," Dec. 11 2007, uS Patent 7,307,357.

[6] R. D. Hippenstiel, *Detection theory: applications and digital signal processing*. CRC Press, 2001.

[7] L. Frenzel, *Principles of electronic communication systems*. McGraw-Hill, Inc., 2007.

[8] N. M. Boers, I. Nikolaidis, and P. Gburzynski, "Impulsive interference avoidance in dense wireless sensor networks," in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2012, pp. 167–180.

[9] W. T. Webb and L. Hanzo, *Modern Quadrature Amplitude Modulation: Principles and applications for fixed and wireless channels: one*. IEEE Press-John Wiley, 1994.

[10] B. E. Rimoldi, "A decomposition approach to cpm," *IEEE Transactions on Information Theory*, vol. 34, no. 2, pp. 260–270, 1988.

[11] M. Uno, "Orthogonal frequency division multiplexing (ofdm) system with channel transfer function prediction," Jun. 10 2008, uS Patent 7,386,072.

[12] Y. Wu and W. Y. Zou, "Orthogonal frequency division multiplexing: A multi-carrier modulation scheme," *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 392–399, 1995.

[13] N. Erdol, F. Bao, and Z. Chen, "Wavelet modulation: a prototype for digital communication systems," in *Proceedings of Southcon'95*. IEEE, 1995, pp. 168–171.

[14] D. Divsalar and M. K. Simon, "Multiple trellis coded modulation (mtcm)," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 410–419, 1988.

[15] R. C. Dixon, *Spread spectrum systems: with commercial applications*. Wiley New York, 1994, vol. 994.

[16] R. Sadr and J. K. Omura, "Generalized minimum shift-keying modulation techniques," *IEEE transactions on communications*, vol. 36, no. 1, pp. 32–40, 1988.

[17] J.-H. Kim, J.-G. Kim, Y.-H. Ji, Y.-C. Jung, and C.-Y. Won, "An islanding detection method for a grid-connected system based on the goertzel algorithm," *IEEE Transactions on Power Electronics*, vol. 26, no. 4, pp. 1049–1055, 2011.

SESSION 4

Text Mining

Lyrics-based Classification of Portuguese Music

David C. T. Freitas
Doctoral Program in Informatics Engineering
FEUP
Porto, Portugal
ec10146@fe.up.pt

***Abstract*—There is a huge amount of music available to everyone, everywhere. Recommendation systems are now an essential part of every music lover activity. Although containing an important part of the information, music lyrics are usually left out of most classification systems. In this paper, we show, just by considering song lyrics, leaving out the digital audio signal, that we can classify its music genre. For this task, we consider two different approaches: cosine similarity and Naïve-Bayes. We also include a way of classifying each song as positive, neutral or negative. This work focuses only on Portuguese songs. The results were positive in both methods making it possible for Portuguese music to be classified based solely only on text lyrics, or at least, to be used in combination with the digital audio signal.**

***Index Terms*—Natural Language Processing, Text Mining, Music classification**

I. Introduction

The expression "music is a universal language", has nowadays a new meaning. There is a huge amount of music-related material available everywhere, much of it free of charge, as normally is the case with lyrics.

Finding relevant music is an important task for music "aficionados" or for applications like Spotify. Spotify has currently more than 30 million songs available to its users. However, the recommendation systems rarely use the lyrics of the songs, focusing essentially on the analysis of the digital audio signal, leaving out an important part of the music. Music lyrics have a great impact on our society, and in our recent history, some songs' lyrics even became symbols of revolutions. Singer-songwriters, like Leonard Cohen or Bob Dylan, have been acknowledged by what they say in their songs. The latter was even awarded the Nobel Prize in Literature in 2016. Taking all those facts into consideration, including the lyrics in the analysis of music classification processes may benefit them.

Lyrics are usually easy to find online and are the preferred way to search for music for non-musicians users [1]. From a cognitive point of view, it must be highlighted that processing the musical lyrics and the melody, is done in an independent way by our brain [2].

In this work, we intend to develop a system allowing, solely from the textual part of the lyrics of the song, classify its musical genre. A sentiment classification of each song is also done. This work has the novelty of being applied to Portuguese music.

II. Related Work

There are many papers on automatic classification of music using lyrics in English, however, for the specific case of music with Portuguese lyrics, no similar work was available. Some implementations use an approach based on n-grams [3], [4], but usually also include more sophisticated features based on different dimensions of a song text, such as vocabulary, style, semantics, and song structure.

Other approaches use Naïve Bayes, to successfully predict song performer based solely on lyrics [5], although in a very simplified environment (two authors and only 207 songs).

TABLE I
Documents in the corpus distributed by genre

Id	Genre	#Docs
1	Romantic	118
2	Fado	94
3	Rap/Hip-hop	104
6	Pimba	83
7	Pop-Rock	101

III. General Considerations

For this project, a database was created using MySQL containing a corpus of 500 songs, distributed by the styles identified in Table I.

All songs in the dataset were downloaded from various sites from the Internet and manually annotated.

For the analysis of the songs we used Python, with the NLTK toolkit [6], and a graphical interface was developed using HTML, PHP, JavaScript, and CSS.

The corpus contains 67,708 words, of which 12,558 are different tokens (unique words). For this study, only words appearing at least 15 times were considered.

The word that appears the most in our corpus is "love" (646 times). Love is everywhere. A list of the 10 most frequent tokens can be found in Table II.

Some words of the corpus were assigned a polarity value (positive, negative, or neutral) and a grammatical class. For the polarity value, SentiLex-PT [7] was used. SentiLex-PT is a sentiment lexicon for Portuguese, made up of 7,014 lemmas, and 82,347 inflected forms. The class of the words should be considered in a posterior version of this application to increase the accuracy of the predicted genre. The polarity of the words is used to predict the polarity of each song lyric and the generality of the corpus.

IV. METHODOLOGY

The development of this project was made according to the CRISP-DM methodology [8]. This methodology is an iterative

TABLE II
THE 10 WORDS WITH GREATEST FREQUENCY

Word	Frequency	Class	Polarity
amor	645	N	
vida	507	V	
tudo	501	PRO	
mim	463	PRO	
ti	424	N	
porque	422	CONJ	
ser	399	V	1
quero	387	V	
vou	384	V	
bem	370	N	

process model for data mining that provides an overview of the life cycle of a data mining project. The pipelining of the process can be seen in Figure 2.

A. *Business understanding / Data understanding*

Understanding the specificity of music classification by genre is crucial to the task of automatic classification. One of the hardest problems to grasp is the ambiguity of each song classification. Different persons can classify the same song to a different genre.

A problem even harder is that the same lyrics can be sung in different styles. For example, "Povo que lavas no rio", has been sung by António Variações and Amália Rodrigues, in two different genres. The former as a Pop/Rock song and the latter, as a Fado song.

Irony can also be hard to track. Songs like "I'm so happy, I can't stop crying" by Sting or "It only makes me laugh" by Oingo Boingo are songs that can be classified as happy but are about sad moments. The polarity of the song could influence the way a song is classified.

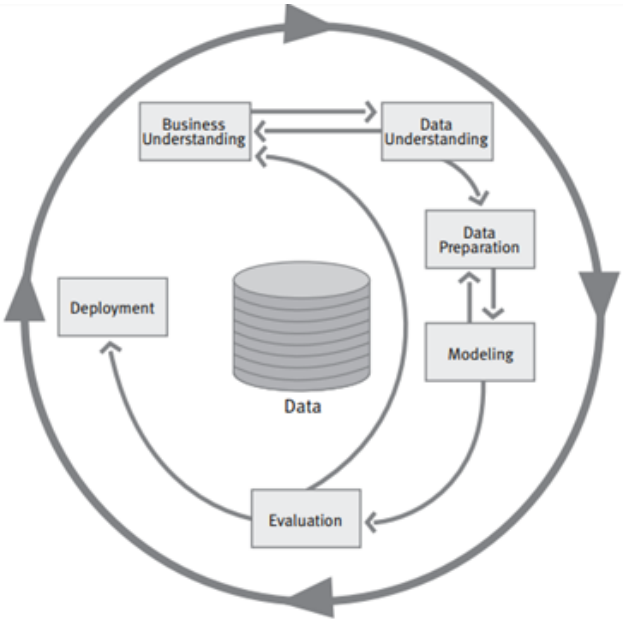


Fig. 1. Phases of the CRISP-DM reference model

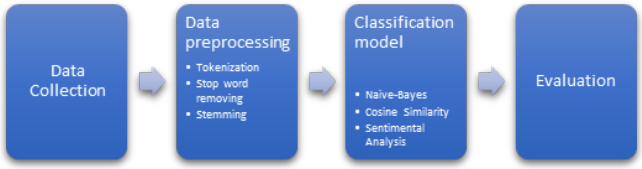


Fig. 2. Pipelining of the project

The mutability of a song during its performance can also make it difficult to classify. For example, the song "Bohemian Rhapsody" by Queen has great variations and can be hard, maybe impossible, to classify with consensus.

Those difficulties are indicators of why lyrics are usually left out of music recommendation systems, but it must be noted that similar problems occur when classifying songs based solely on digital audio signal. Singer-songwriters like Frank Zappa were masters in using elements from one music style to create a song in a totally different genre.

B. *Data preparation*

A great amount of effort and time was spent preparing the data.

After the tokenization of each lyric, some words were discarded. Portuguese stopwords ('a', 'ao', 'aos', 'aquela', 'aquelas', 'aquele', 'aqueles', ...), punctuation and isolated letters were removed.

Most of the documents were retrieved from the Internet and contained strings like "Instrumental", "Refrão" or (x2) that needed to be removed from the tokenized words. Some problems with codification were also considered. Strings like "\ufeff" and files not saved in "utf-8" needed to be converted. Regular expressions were used to find patterns like "aaaaah", "oooooh" or symbols that for no apparent reason appeared in the lyrics ("———", "######", ...).

C. *Modeling*

The goal was to implement a classifier able to identify a genre of a song amongst the available classes: "Romântica", "Fado", "Rap/Hip-Hop", and "Pimba". This type of problem is known as a multiclass classification problem.

Two different models were used to classify each song contained in our test lyric: **Cosine similarity** and **Naïve Bayes**.

In both models, during the analysis of the corpus, the set of all words considered is denotated by W.

For each song in the database, a **polarity value** was attributed (sentimental analysis).

Polarity analysis

Each song in the database was attributed a positive, neutral or negative classification. Although this approach was rude, consisting only of counting the more frequently polarity value (positive, neutral, or negative), but effective. It must be noted that this process was only evaluated empirically. The validation of this process should be further developed.

Cosine similarity

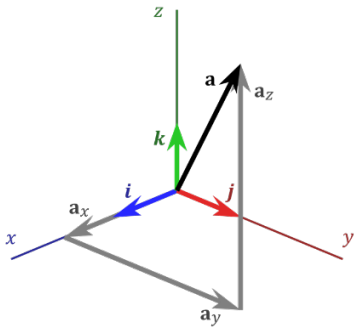


Fig. 3. Every token can be seen as a dimension of the hyperspace

A vector space model (VSM) was implemented to represent the lyrical data. Each document in our training set is represented by a vector $\overrightarrow{w}=(w_1,w_2,\ldots,w_n)$, where if $w\in W$ is a token word considered in the corpus, then there is one and only one i: $1\leq i\leq n$, such that $w_i=w$ and n is the size of the corpus. In this particular case of study, the number of words considered was n=799. We remember that this number was obtained considering all the words appearing more than 15 times and were not discarded in the process of removing the stop words or words considered irrelevant.

Every document d_i in the corpus was transformed into a vector $\overrightarrow{d_i}=(f_{i1},f_{i2},\ldots,f_{in})$, where f_{ij} is the frequency of the word w_j in the document i, where $w_j\in W$. This representation consists of a sparse term frequency matrix of size $n\times 799$, where $n=500$ songs in the dataset.

Each document can be considered as a vector in hyperspace where the frequency of every word considered in the corpus is represented in one dimension (Figure 3). For every style S, the following vector was obtained:

$$\overrightarrow{s}=\sum_{d\in S} d_i \tag{1}$$

The Cosine Similarity between two vectors x and y can be expressed by the following formula:

$$sim\,(x,y)=\frac{x\bullet y}{\|x\|\,\|y\|} \tag{2}$$

where $\|x\|$ and $\|y\|$ are the Euclidean norm for the vectors x and y.

For every document $\overrightarrow{t}$ in the test list, the maximum value of $sim\left(\overrightarrow{t},\overrightarrow{s}\right)$ allowed to choose the attributed style.

Considering that a document with many occurrences of the same word doesn't necessarily increase proportionally its importance and given the fact that a long document has a higher probability of having more words, the following correction was applied [9]. This approach is similar to the one used in the tf-idf weighting. For each vector representing style s.

$$s=round\left(10\times\frac{1+\log\left(f_n\right)}{1+\log\left(l\right)}\right) \tag{3}$$

TABLE III
EVALUATION OF THE SYSTEM

Genre	Naïve-Bayes	Cosine similarity
Romântica	15 / 20 (75%)	9 (45%)
Fado	16 (80%)	18 (90%)
Rap/Hip-Hop	14 (70%)	18 (90%)
Pimba	11 (55%)	9 (45%)
Total	**56 (70%)**	**54 (67,5%)**

where f_n is the frequency of word n, and l is the length of document d.

Naïve Bayes

In this approach, each new instance of a lyric is classified as a class from the set S. To estimate the probability of a lyric D, belong to the class C, we applied Bayes' theorem [10]:

$$P\left(C\mid D\right)=\frac{P(D|C)P(C)}{P(D)} \tag{4}$$

Assuming the probability of two words appearing in the same document are independent, which we know isn't totally true but can be used to simplify the equation, we obtain:

$$P\left(C\mid D\right)=P(f_{i1}|C)\times P(\ f_{i2}|C)\times P(f_{in}|C)\times P(C) \tag{5}$$

To predict the class label, we select the genre with the maximum probability.

D. Evaluation

To evaluate the results, we considered 80 new songs (20 songs for every style, "Pop/Rock" was not tested in this first approach because of its high variability), that weren't available in the training of both models.

The precision of the system can be seen on Table III.

E. Deployment

An interface was developed to help how to understand which information is relevant to the analysis of the classification. During the processing of each song, a JSON file with information about it is generated.

A resume of the corpus is made available and can be used to understand which words are important. Polarity of each word, when available, can also be seen on Figure 4. We can also see that although 2.897 words in the corpus had a polarity value, removing the words appearing less than 15 times, only 230 words with polarity value were considered.

For each song, we can highlight what words are more relevant (have more weight) for the analysis, as can be seen in Figure 5.

For each document, the probability of belonging to all considered genres was computed and saved. It is possible to check in which order the correct classification was calculated in Figure 6. For examples, the song "O ideal", was correctly classified as the 3rd probable genre by Naïve-Bayes approach, while correctly identified as the 1st probable genre by Cosine-Similarity. The song "Retratos de uma cidade branca", was identified as last probable genre, by both methods. These worst classified songs can be useful to analyze what went wrong in the process.

Corpus

Corpus size	799
Minimum words	15
Total tokens	12558
Total words	67708
Total tokens deleted	11759 (93.64%)
Total words deleted	28232 (41.7%)
Number of words with polarity	230 (2897)
Corpus polarity	16 (-313)

Word	Frequency	Class	Polarity
amor	646	N	
vida	507	V	
tudo	501	PRO	
mim	463	PRO	
ti	424	N	
porque	422	CONJ	
ser	399	V	1
quero	387	V	
vou	384	V	

Fig. 4. Resume of the corpus of the documents in study

Fig. 5. Words relevant for a particular lyric

V. FUTURE WORK

Many things could be done to improve the automatic classification of lyrics. POS tagging applied to the words appearing in our corpus should be considered in the classification of the lyrics. N-grams could also be used to improve the system. The length of the sentences and structure of a song (chorus, verse, ...), although usually hard to detect in a "lyric format" could be useful to the process.

Some new services like finding the date of the publication, identifying the mood of the song or identifying the writer could be retrieved from the database. Some authors use a unique style for their lyrics and, surely some characteristics, could be obtained automatically, allowing its identification.

Fig. 6. Correct order of classification of each lyric

VI. CONCLUSIONS

This paper shows that a song can be classified by its genre, with good accuracy, considering only its lyrics. Extraction of useful information for musicologists can certainly be possible. Analysis of lyrics could be very helpful to use as a proxy for other structures, like the chorus, verse or bridge.

The results obtained by both approaches, Naïves-Bayes classification, and the Vector Space Model performed approximately with the same success. Some musical genres seem to have better signatures to be identified, like "Fado" or "Rap/Hip-Hop" but more experiences should be done to confirm these results.

REFERENCES

[1] Baumann, S., Kluter, A., and Fingerhut, M.: 'Super-convenience for non-musicians: querying MP3 and the Semantic Web', ISMIR 2002 Conference Proceedings. (Third International Conference on Music Information Retrieval), 2002, pp. 297-298
[2] Besson, M., Faïta, F., Peretz, I., Bonnel, A.M., and Requin, J.: 'Singing in the Brain: Independence of Lyrics and Tunes', Psychological Science, 1998, 9, (6), pp. 494-498
[3] Michael Fell.: 'Lyrics-based Analysis and Classification of Music' (2014)
[4] Mahedero, J.P.G., Cano, P., Koppenberger, M., and Gouyon, F.: 'Natural language processing of lyrics'. Proc. Proceedings of the 13th annual ACM international conference on Multimedia, Hilton, Singapore 2005
[5] Bužić, D., and Dobša, J.: 'Lyrics classification using Naïve Bayes': 'Book Lyrics classification using Naïve Bayes' (2018, edn.), pp. 1011-1015
[6] Loper, E., and Bird, S.: 'NLTK: the Natural Language Toolkit'. Proc. Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1, Philadelphia, Pennsylvania 2002
[7] Ranchhod: 'SentiLex-PT 02', in Editor 'Book SentiLex-PT 02' (2012).
[8] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., and Wirth, R.: 'CRISP-DM 1.0 Step-by-step data mining guide', in Editor 'Book CRISP-DM 1.0 Step-by-step data mining guide' (2000).
[9] Manning, C.D.: 'Foundations of statistical natural language processing' (MIT Press, 1999)
[10] Han, J., Kamber, M., and Pei, J.: 'Data Mining: Concepts and Techniques' (Morgan Kaufmann Publishers Inc., 2011)

An Application of Information Extraction for Bioprocess Identification in Biomedical Texts

Paula Raissa Silva
FEUP, INESC TEC, Portugal
up201802218@fe.up.pt

***Abstract*—Nowadays it was observed a large diversity of electronic publications in scientific databases. So, the biomedical researches spend a lot of time and effort in searching for available information. This problem is caused by the fact that there are various name expressions for the same biological subject, orthographic variants and abbreviations. In this case the existing search engines can not deal with this complexity. In this paper we propose an NLP-based approach integrating NLP techniques and decision tree, for extracting biomedical events from scientific literature, utilizing the corpus from the BioNLP'16 Shared Task on Protein Regulation. The experimental results show that the presented approach can achieve an F-score of 0.57 in the test set, which reaches the same result of 3 state-of-the-art official submissions to BioNLP 2018. So, the presented approach demonstrate the potential to help the organization of electronic publication according to the biological subjects inside the scientific articles.**

***Index Terms*—Biomedical events, text mining, bionlp, spacy, random forest**

I. INTRODUCTION

Biomedical science is an area where the most part of knowledge is represented as free text, which could be splitted into two main groups: clinical texts and scientific text. The clinical texts generally are produced by scientists and doctors about their observations, producing an unstructured information making it difficult to identify relevant knowledge. The scientific text group is composed by articles and any scientific publication, which the knowledge is represented through free-text with some formalism. It was observed that in both cases, clinical and scientific texts, the problem with the large amounts of documents and the time spent by researchers to find relevant informations about the objects and events that they are studying. So, the application of automatic information extraction and knowledge identification techniques can help the researchers and doctors to reduce the time spent in the literature review process.

Biomedical Text Mining is a promising research area, that deals with automatic retrieval and processing of biomedical texts [1]. There are two most known groups of approaches: rule-based or knowledge-based, and statistical or machine learning based approaches. The common tasks in BTM include Named Entity Recognition (NER), Relation Extraction, document summarization, classification and clustering. Supporting and motivating more researches to contribute in this area, it was created the BioNLP Shared Task in 2009 that has been promoting the development of fine-grained information extraction from biomedical documents. Every two years, this group elect three or fours tasks around different biomedical problems and promote a competition with the aim of produce the state of the art for biomedical text mining applications.

Recently, machine learning methods provide an effective way to automatically extract relevant knowledge and achieve notable results in various NLP tasks. So, in this work, we present an approach that combines NLP techniques and decision trees to identify biomedical events in scientific texts utilizing the corpus from the BioNLP'16 Shared Task on Protein Regulation [1].

The structure of the paper follows as: In Section 2, we present the definitions of the main concepts addressed throughout the work. In Section 3, we report related work that applied text mining to identify biological events. In Section 4, we present the used materials and the pipeline. In Section 5, we report experiments and results. Finally, in Section 6, we present the conclusions and future work.

II. BACKGROUND

Automatic biomedical event extraction is a multidisciplinary task and represents a contribution to the progress of biomedical domain. In this section we introduce the main theoretical reference covered by this work.

A. Biological Events

By definition, a biological event is any vital process executed by organisms [2]. Generally, a biological event is a recognized series of molecular functions. There is some relevant event types like gene expression, transcription, protein catabolism, phosphorylation, localization, binding and regulation.

B. Parsing

Parsing in NLP (Natural Language Processing) is the process of determining the syntactic structure of a text by analyzing its constituent words based on a grammar. The outcome of the parsing process would be a parse tree, where sentence is root, intermediate nodes as noun_phrase, verv_phrase are called non-terminals and the leaves are called terminals. Existing parsing approaches ares basically statistical, probabilistics and machine learning-based. There are four types of parsing:

- Shallow parsing (chunking): It is based on hierarchy based on groups of words make up phrases [3]. The most common operation is grouping words into Noun Phrases (NP). But there is other categories like Verb Phrase (VP),

[1] http://2016.bionlp-st.org/tasks/ge4

Adjective Phrase (ADJP), Adverb Phrase (ADVP) and Prepositional Phrase (PP). Example: Genia Tagger[2].

- Constituency parsing (deep parsing): This type of parsing is used to analyze and determine the constituents of a sentence according to constituent-based grammars [4]. These grammar can be used to model or represent the internal structure of sentences following an ordered structure of their constituents. Each word belongs to a specific lexical category in the case and forms the head word of different phrases. Finally the phrases are formed based on rules called structure rules. Example: Stanford Parser[3].
- Dependency parsing: It implies in find the dependencies between the words and their type. According to [5] the parser needs a dependency based grammars to analyze and infer the structure and semantic dependencies and relationships between tokens in a sentence. The basic principle is that any sentence in a language, all words except one, have some relationship or dependency on others words in the sentence. The word that has no dependency is called root. The verb is taken as the root of the sentence. All the other words are directly or indirectly linked to the root verb by links.
- Part Of Speech (POS) Tagging: Parts of speech are specific lexical categories to which words are assigned, based on their syntactic context and role [6]. Example: NLTK[4] and Spacy[5].

C. Random Forests

The decision tree classifier is a supervised learning algorithm which can use for both the classification and regression tasks [7]. The most important characteristic of decision trees is the capability to lead with complex decision making problems into a collection of simpler decisions, and provide a solution in a easy way to interpret [8]. Random Forests are a combination of decision trees which each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. In the last years, this machine learning method showed itself capable of performing both regression and classification problems. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data analysis [9].

Random Forests build a number of decision trees on bootstrapped training samples. But when building these decision trees, each time a split in a tree is considered, a random sample of m predictors is chosen as split candidates from the full set of p predictors. The split is allowed to use only one of those m predictors.

Following can be observed an pseudocode in order to grow a random forest:

- First assume that the number of cases in the training set is K. Then, take a random sample of these K cases, and then use this sample as the training set for growing the tree.
- If there are p input variables, specify a number $m < p$ such that at each node, you can select m random variables out of the p. The best split on these m is used to split the node.
- Each tree is subsequently grown to the largest extent possible and no pruning is needed.
- Finally, aggregate the predictions of the target trees to predict new data.

Random Forests is effective at estimating missing data and maintaining accuracy when a large proportions of the data is missing. It can also balance errors in datasets where the classes are imbalanced. Most importantly, it can handle massive datasets with large dimensionality. However, the disadvantage of using Random Forests is that it easily overfit noisy datasets, especially in regression applications.

III. RELATED WORKS

Actually several authors have been worked in the issue of biomedical event extraction. Most of them is contributing to BioNLP Shared Task [10], [11], [12], [13]. The BioNLP Shared Task has been organized since 2009 and the goal is to provide the community with shared resources for the development and evaluation of biomedical information extraction systems, mainly for the domain of molecular biology and medicine [14].

Recently, supervised learning methods have provided an effective way to automatically extract features and achieve notable results in various natural language processing tasks. The authors have used Hidden Markov Models [15], decision trees [16], support vector machines [17], deep learning [18] and conditional random fields [19]. It was also observed applications that combine unsupervised and supervised machine learning approaches, introducing a supervised approach trained on the combination of trigger word classification produced by the unsupervised clustering method and manual annotation [20].

In the state of the art, it was also identified tools for event extraction based in dependency parsing. TrigNER is a event trigger recognition, based in dependency parsing, that uses an optimization algorithm which allows the tool to adapt itself to corpora with different events and domains [21]. The approach provides a new insight on the linguistic and context complexity of each event trigger and associated requirements. The feature set are based on concept-based features like:

- Tag: if a token is part of concept name, the tag is added to the token associated feature.
- Name: name of the concept is a part of.
- Head: Head token of concept name.
- Counting: number of associations per concept type.
- Dictionary: dictionaries os specific domain terms and trigger words are used as features.

Turku is an event extraction system based in the analysis of the dependency parse correlation with annotations. According

[2]http://www.nactem.ac.uk/GENIA/tagger/
[3]https://nlp.stanford.edu/software/lex-parser.shtml
[4]https://www.nltk.org/
[5]https://spacy.io/

to [22] the shortest path is the primary source of features for interactions, where they are binary relations or event arguments. It applies the Porter Stemmer [23] to make the derivation of stem for each word. The stem and non-stem words are used as features to detect the same word in different inflected forms.

IV. MATERIALS AND METHODS

In this section will be explained the main materials used to build this project, like the BioNLP database and the Spacy parser. It also shows the builded pipeline and each process executed inside the pipeline steps.

A. The Database

The database was extracted from BioNLP organization, and it is basically composed by scientific articles in biomedical area. It is composed by 471 *json* files, where each file represent one article structures in target, sourcedb, sourceid, text, project, denotations and relations. Splitting the text into sentences and denotations, it could be counted 4612 sentences and 43236 denotations. It is important to define that the denotation is the type of annotation that helps the identification of events and triggers.

The Figure 1 shows an example of the structure of *json* files in the BioNLP Shared Task dataset[6]:


```
{
"target":"http://pubannotation.org/docs/sourcedb/PMC/sourceid/1134658/divs/0",
"sourcedb":"PMC",
"sourceid":"1134658",
"divid":0,
"text":"BMP-6 inhibits growth of mature human ..."
"denotations":[{"id":"T1","span":{"begin":0,"end":5},"obj":"Protein"},...}],
"relations":[{"id":"R1","pred":"themeOf","subj":"T2","obj":"E1"}...],
"modifications":[{"id":"M1","pred":"Negation","obj":"E15"}...],
"namespaces":[{"prefix":"_base","uri":"http://bionlp.dbcls.jp/ontology/ge.owl#"}
]}
```


Fig. 1. Example of BioNLP json file.

In this project each denotation object was considered as a class. So, the possible classes are: acetylation, binding, DNA, deacetylation, entity, gene expression, localization, negative regulation, phosphorylation, positive regulation, protein, protein catabolism, protein domain, protein modification, regulation, transcription and ubiquitination. However, some of these classes have few objects to be able to train a model. So these classes with support less than two was removed, for example DNA with 1 object, Acetylation with 2 objects, and others.

B. Spacy

In this work, the Spacy was applied to extract linguistic features like POS tags. After tokenization, Spacy can parse and tag a given document. This document can be a sentence or a group of sentences readed by the function NLP. For the POS functionality, Spacy runs the statistical model that makes a prediction of which label most likely applies in the context. The model is a binary data and is produced by testing the system with examples for make predictions generalising across the language. For example, a word following "the" in English is most likely a noun. Spacy encodes all strings to hash values to reduce memory usage and improve efficiency. So, we extracted the following features:

- Text: the token.
- Lemma: base form of the word.
- POS: simple POS tag, like NOUN, VERB, ADP, etc.
- Tag: detailed part of speech tag.
- Dep: relation between tokens.
- Alpha: define if the token is an alpha character.
- Stop: define if the token is the most common word of the language.

C. The Pipeline

To visualize in a simple way, the pipeline was splitted in four steps and was showed in the Figure 2.

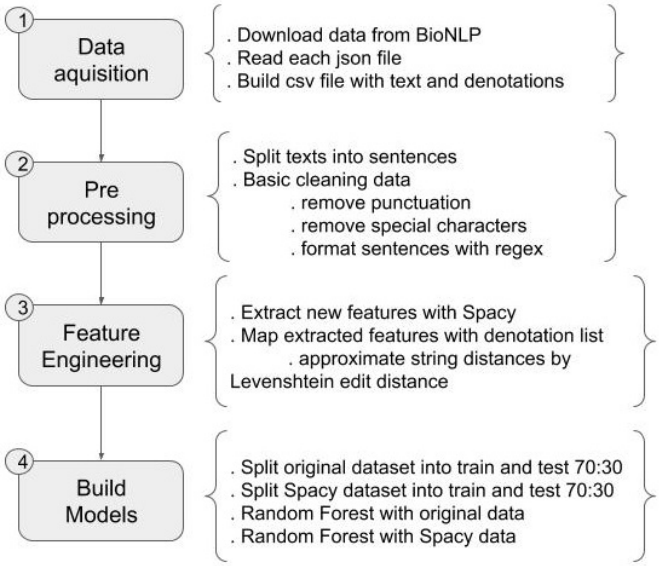


Fig. 2. The Pipeline

V. EXPERIMENTS

This section describes the results achieved. The experiments was built around the research question: Does adding new features improve the classification result?

Aiming to answer the research question, firstly we trained the model using only the text feature and the label. Is important to note that all the classification models was built using the *CaretR* package. So, in this case we could generate the results showed in Table I.

After, another model was trained using the spacy features and the label. The features extracted with the spacy parser was $token.tag$, $token.dep$, $token.is_alpha$, $token.is_stop$ and n (the number of tokens by class). So, the Table II shows the results for this classification model. Analysing this table, it is already possible to check that the spacy features did not help the classifier to improve the model, because the precision, recall and f-score are less than the previous model.

[6]Available at: http://2016.bionlp-st.org/tasks/ge4

	Precision	Recall	F-Score
Binding	0.57	1.0	0.73
Entity	0.28	1.0	0.44
Gene_expression	0.39	1.0	0.57
Localization	0.33	1.0	0.50
Negative_regulation	0.31	1.0	0.47
Positive_regulation	0.49	1.0	0.46
Protein	1.0	0.57	0.72
Regulation	0.31	1.0	0.48
Transcription	0.58	1.0	0.74

TABLE I
METRICS FOR TEXT CLASSIFICATION

	Precision	Recall	F-Score
Binding	0	0	0
Entity	0	0	0
Gene_expression	0	0	0
Localization	0	0	0
Negative_regulation	0.25	0.34	0.29
Positive_regulation	0.43	0.33	0.37
Protein	0.94	0.58	0.71
Regulation	0	0	0
Transcription	0	0	0

TABLE II
METRICS FOR SPACY CLASSIFICATION

Finally it was compared the both results given by the random forest algorithm. To make easy the results comparison, it was considered only the F-score measure because it is a combination between precision and recall indicating the quality of the model [24].

In Table III can be observed the comparison between the classification results using text and using text plus spacy features. According to this results the classification using the spacy features works worse than the classification test using only the text variable. Exploring the data, it was observed a relation between the class support and F-Score measure. In the case of Protein class, with the highest support (410), there is no significant difference between the models (0.01). For the classes *negativeregulation* and *positiveregulation* the inclusion of new features worsened the classification. And, for the classes *binding*, *entity*, *geneexpression*, *localization*, *regulation* and *transcription* the model could not associate any instance. The reason for this behavior is still under review, because we are expecting better results when adding new features.

	F-Score Text	F-Score Spacy
Binding	0.73	0
Entity	0.44	0
Gene_expression	0.57	0
Localization	0.50	0
Negative_regulation	0.47	0.29
Positive_regulation	0.46	0.37
Protein	0.72	0.71
Regulation	0.48	0
Transcription	0.74	0

TABLE III
TEXT VERSUS SPACY FEATURES

VI. CONCLUSIONS

Finally, this section talks about the results discussion and suggestions for improvements and future works.

In general, the Random Forest algorithm achieve good results, even this is not the best algorithm to work with high dimensionality. But the results is still preliminary, because to prove if they are valid the pipeline needs to be executed more than once and the Classification step need to be tested with another algorithms and another samples of train and test with different sizes. Another important improvement is make the tuning of parameters aiming to find the best configuration for building the models.

According to the results, the features extracted by Spacy POS tagging could not improve the classification results, because in this case adding information to the classifier did not help the algorithm to map objects for the right classes. In this case, for future works, it can be measured how many and what features really helps the classifier in the mapping process, because according to the literature there is a limit of information that can be added in the feature set resulting in good classification measures.

REFERENCES

[1] R. Rodriguez-Esteban, "Biomedical text mining and its applications," *PLoS computational biology*, vol. 5, no. 12, p. e1000597, 2009.

[2] M. J. Meaney, "Epigenetics and the biological definition of gene× environment interactions," *Child development*, vol. 81, no. 1, pp. 41–79, 2010.

[3] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp. 134–141, Association for Computational Linguistics, 2003.

[4] D. McClosky, "Any domain parsing: automatic domain adaptation for natural language parsing," 2010.

[5] M.-C. De Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, and C. Manning, "Universal stanford dependencies: A cross-linguistic typology," *Proceedings of the 9Th International Conference on Language Resources and Evaluation (LREC)*, pp. 4585–4592, 01 2014.

[6] L. Màrquez and H. Rodríguez, "Part-of-speech tagging using decision trees," in *European Conference on Machine Learning*, pp. 25–36, Springer, 1998.

[7] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[8] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[9] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] J. Wang, Q. Xu, H. Lin, Z. Yang, and Y. Li, "Semi-supervised method for biomedical event extraction," *Proteome Science*, vol. 11, p. S17, Nov 2013.

[11] H. Liu, K. Verspoor, D. C. Comeau, A. D. MacKinlay, and W. J. Wilbur, "Optimizing graph-based patterns to extract biomedical events from the literature," *BMC Bioinformatics*, vol. 16, p. S2, Oct 2015.

[12] S.-C. Baek and J. C. Park, "Making adjustments to event annotations for improved biological event extraction," *Journal of Biomedical Semantics*, vol. 7, p. 55, Sep 2016.

[13] L. Li, J. Wan, J. Zheng, and J. Wang, "Biomedical event extraction based on gru integrating attention mechanism," *BMC Bioinformatics*, vol. 19, p. 285, Aug 2018.

[14] J.-D. Kim, J.-j. Kim, X. Han, and D. Rebholz-Schuhmann, "Extending the evaluation of genia event task toward knowledge base construction and comparison to gene regulation ontology task," *BMC Bioinformatics*, vol. 16, p. S3, Jun 2015.

[15] S. Zhao, "Named entity recognition in biomedical texts using an hmm model," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pp. 84–87, Association for Computational Linguistics, 2004.

[16] S. Sekine, "Nyu: Description of the japanese ne system used for met-2," in *Proc. of the Seventh Message Understanding Conference (MUC-7*, Citeseer, 1998.

[17] K.-J. Lee, Y.-S. Hwang, and H.-C. Rim, "Two-phase biomedical ne recognition based on svms," in *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine-Volume 13*, pp. 33–40, Association for Computational Linguistics, 2003.

[18] H.-J. Song, B.-C. Jo, C.-Y. Park, J.-D. Kim, and Y.-S. Kim, "Comparison of named entity recognition methodologies in biomedical documents," *Biomedical engineering online*, vol. 17, no. 2, p. 158, 2018.

[19] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pp. 188–191, Association for Computational Linguistics, 2003.

[20] F. Mehryary, S. Kaewphan, K. Hakala, and F. Ginter, "Filtering large-scale event collections using a combination of supervised and unsupervised learning for event trigger classification," *Journal of biomedical semantics*, vol. 7, no. 1, p. 27, 2016.

[21] D. Campos, Q.-C. Bui, S. Matos, and J. L. Oliveira, "Trigner: automatically optimized biomedical event trigger recognition on scientific documents," *Source Code for Biology and Medicine*, vol. 9, p. 1, Jan 2014.

[22] J. Björne *et al.*, "Biomedical event extraction with machine learning," 2014.

[23] P. Willett, "The porter stemming algorithm: then and now," *Program*, vol. 40, no. 3, pp. 219–223, 2006.

[24] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation," in *Australasian joint conference on artificial intelligence*, pp. 1015–1021, Springer, 2006.

Natural Language Analysis of Github Issues

Fávio Henrique Ferreira Couto
Faculty of Engineering
University of Porto
Porto, Portugal
up201303726@fe.up.pt

***Abstract*—Software development can be simplified as continuous series of tasks of translating requirements into program features. One source of said requirements is the issue tracker of deployed systems. In it, users are able to provide developers with descriptions of their problems. However, this system can get difficult to manage and reports on the same issue are common to appear. By utilizing Natural Language analysis methods, we develop and evaluate a system able to identify key elements, propose labels, list similar issues and, to a smaller extent, predict the time needed to close an issue.**

***Index Terms*—Text Mining, Natural Language Processing, Requirement Analysis**

I. Introduction

Software development is a complex process for which multiple methodologies were developed since its beginnings. However, we can also view it simply as the process of translating requirements into executable programs. One of the biggest challenges for this process is that the requirements are usually provided in Natural Language and the programs are developed with a more strict, machine-related language. Despite efforts from methodologies such as Behavior-Driven Development [1], that aim to bridge this gap, the analysis of Natural Language requirements is and will continue to be essential for software development.

The requirements of a software project can have many sources, from clients to developers, from the initial idea to the bug reports and feature requests. We selected the Github repository issues as our source of requirements and define a methodology in order to automate some of the analysis of these issues.

II. Methodology

This project follows the CRISP-DM methodology [2], an open standard for data-mining process models. It defines 6 major phases, whose implementation in this project is described in the following subsections and is portrayed in Figure 1.

A. Business Understanding

Github [3] is an online platform providing a Version Control System. For each project repository, developers are able to version control their code, administrate contributors, report issues and use other useful project management tools. Our main goal is to provide relevant data for the developers from the text available in the issues of a repository, in order to facilitate their analysis and, consequently, the development of a solution. In more concrete terms, our objective is to provide

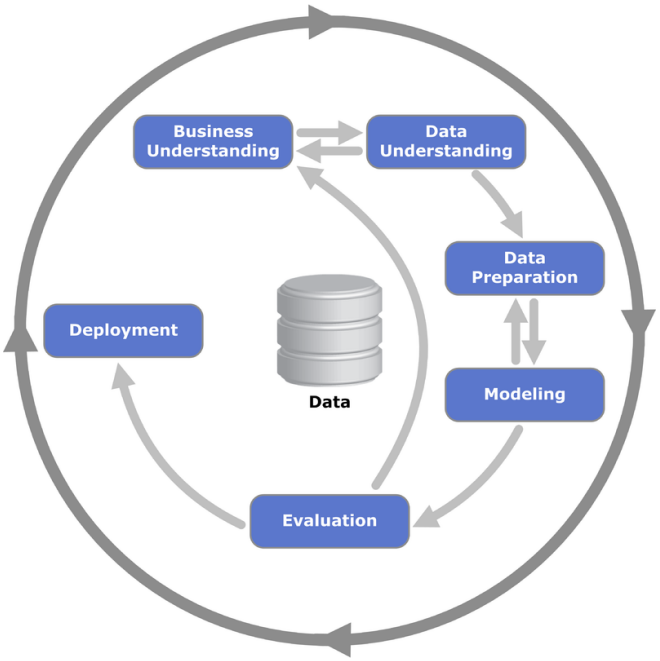


Fig. 1. Process diagram showing the relationship between the different phases of CRISP-DM

a short insight on the issue, containing the four following components:

- Detection of version and issue cross-referencing.
- Suggestion of related labels.
- Listing similar issues.
- Prediction of the time needed to close the issue.

The repository under analysis is titled "Terasology" [4] and contains the source code of a mature project created in 2011 with over 9000 commits and 1500 total issues.

B. Data Understanding

The focal data on this project are the issues on a Github repository. Github provides an API [5] which allows for an easy collection of the issues from a repository. For each issue, we are able to inspect, among other things, title, body, labels, state, author, comments and related dates. The textual data is present in the title and body, describing the issue presented by the other, and are written in English. The labels are mostly used to group issues by their related topics or development status.

TABLE I
DISTRIBUTION OF CODE SECTION RELATED LABELS AMONG THE ISSUES

Label	Number of Issues
Rendering	93
Physics	1
UI	213
Multiplayer	85
Content	119
Artwork	46

In this project, the issues used for model training are the ones reported as closed by their status info, obtainable through a GET request to the URL https://api.github.com/repos/MovingBlocks/Terasology/issues?state=closed . From this resource we collect 1266 documents.

From this process we obtain a record for each document, containing the fields:

- Title: identification / brief summary of the issue.
- Body: detailed explanation of the issue, containing elements such as version number on which the issue occurred and references to other issues.
- Labels: list of labels assigned to the issue.
- Creation Date: date when the issue was created.
- Closing Date: date when the issue was closed.

The repository contributors define 6 code section related labels: *Rendering*, *Physics*, *UI*, *Multiplayer*, *Content*, and *Artwork*. For the collected data, these labels were assigned as portrayed in Table I, with a total of 794 issues without label, 391 with one label and 81 with more than one label.

C. Data Preparation

For the data preparation, a POS tagger was employed, subject to the following rules:

- Tokens that match the regular expression "v\d+.\d+.\d+" are tagged with "VERSION".
- Tokens that match the regular expression "#\d+" are tagged with "ISSUE".
- Other tokens are tagged with the default English POS tagger provided by the NLTK [6].

As a result of this phase, the following fields were added to each record:

- Duration: time needed to close the issue, computed as the days of difference between the creation and closing of the issue.
- Text: Concatenation of the title and body text.
- POS Tags: POS tags assigned to each word of the text with the previously described tagger.
- TF-IDF vector: vector representation of the text obtained from a TF-IDF vectorization of the full collection of documents, with stemming and stopword removal applied.

D. Modeling

According to the objectives defined in Section II-A, four models were developed to fulfill the four tasks.

1) Detection of version and issue cross-referencing: The POS tagging process occurring in the Data Preparation phase renders this task trivial. Both version and issue cross-referencing tokens are detected by their corresponding tag of "VERSION" and "ISSUE".

2) Suggestion of related labels: Due to being able to be assigned multiple labels, their assignment to issues can be interpreted as a Multilabel Classification problem. The labels considered in this classification are the code section related labels mentioned in the section II-A. The classifier for each label is built as One Vs Rest Classifier over a Logistic Regression. Each classifier is trained with 90% of the available documents.

3) Listing similar issues: To find issues similar to the one under analysis, we create a ranking of the other issues ordered by their cosine similarity of their TF-IDF vectors. In order to compute this value, we follow this pipeline:

1) Removal of stop words and stemming;
2) TF-IDF Vectorization of the documents (including the one under analysis);
3) Computation of cosine similarity, as presented in Equation 1;
4) Ordering of issues by the computed cosine similarity value.

$$cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{y}||} \quad (1)$$

4) Prediction of the time needed to close the issue: The prediction of the duration is based upon the text present in the title and body of the issue. This task is modeled as a linear classification problem, whose features are the tf-idf vector of issue and the target value is its duration in days. This problem is tackled with a Gradient Descent Classifier. The model is trained with all of the available documents.

E. Evaluation

Since the **detection of version and issue cross-referencing** and **listing similar issues** were straightforward processes with no tuning parameters to be analyzed, they were only subject to unit case testing with sample issues. For the model tasked with the **suggestion of related labels**, accuracy and f1-scores were computed for the remaining 10% of the documents, not used in the training process. For the model tasked with the **Prediction of the time needed to close the issue**, the R-square metric was computed between the original durations and the predicted ones. As a comparison measure, a more simple classifier that takes into account only the number of words in the body, the presence of a version and the number of issue cross-references was created and its R-square was also computed.

F. Deployment

The models defined in Section II-D are integrated in a bot that analyses the open issues of a Github repository and comments on them with a small report, following the pipeline defined in Figure 2.The report provides a summary of the information produced by each model on a comment of the issue, containing the information about referenced version and

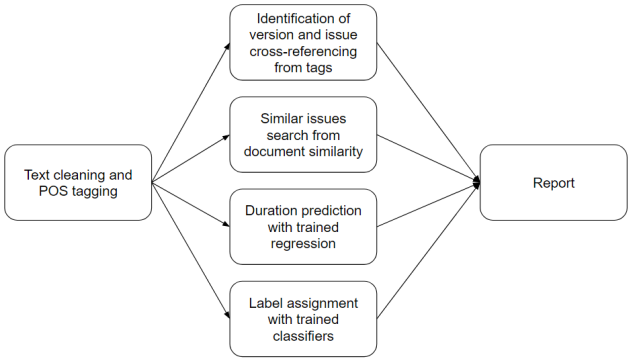


Fig. 2. Pipeline used to analyze a new issue.

TABLE II
METRICS OF THE CLASSIFIER TASKED WITH **SUGGESTION OF RELATED LABELS**

Label	Accuracy	F1-Score
Rendering	81.41%	0%
Physics	100%	0%
UI	82.69%	79.07%
Multiplayer	82.05%	12.50%
Content	75.64%	20.83%
Artwork	89.74%	20%

related issues in the text, a proposal of labels to assign to the issue, a listing of issues that can be related to it and a prediction of the days needed to close the issue. As an example, this is the data (title and body) and info produced in an analysis of an issue:

- Title: "Enabling Animated Menus breaks widget tabbing"
- Body: "When animated menus are enabled, the widget tabbing will show strange activity in the main menu:\n* On the main menu, the widget tabbing works.\n* When an animation forward to any menu occurs the first time, widget tabbing doesn't work and returns an error in console.\n* Any other forward animation after that will result in the tabbing not working without an error in console.\n* But, when a backwards animation occurs, the widget tabbing works again!\n This issue was observed on v1.5.3."
- Version: v1.5.3
- Cross-referenced issues: None
- Proposed labels: UI
- Related issues: #2695, #4, #2762
- Predicted duration: 458 day(s)

III. Results and Discussion

For the **suggestion of related labels**, the classifiers produced the accuracy and f1-scores portrayed in Table II.

As we are able to grasp from these metrics, the UI classifier is the most reliable of them. On the other hand, the Rendering and Physics classifier both have an F1-Score of 0%. This can be explained from the low number of documents with these

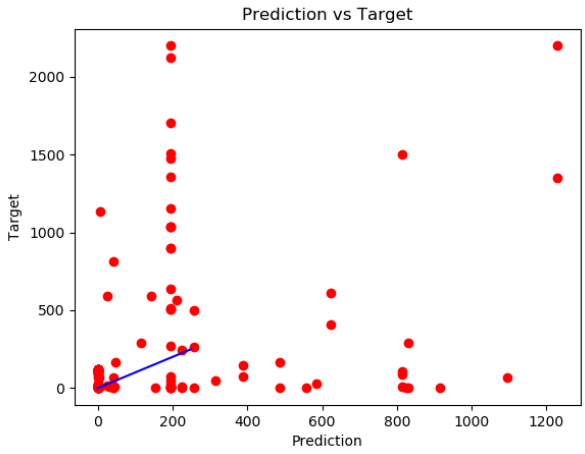


Fig. 3. Comparison of the real and the predicted duration days of issues by the basis classifier. An optimal classifier would result in a linear relation between the data points.

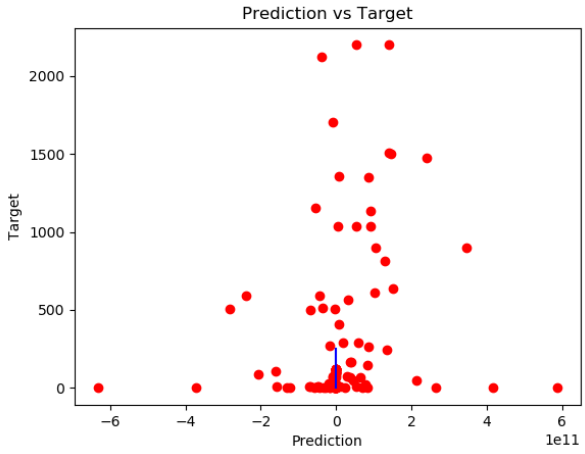


Fig. 4. Comparison of the real and the predicted duration days of issues by the natural language-based classifier. An optimal classifier would result in a linear relation between the data points.

labels, needing more training data in order to be considered useful.

For the model tasked with the **Prediction of the time needed to close the issue**, the basis classifier obtained performance can be seen in Figure 3 an R-square of 0.31%. Our developed classifier is able to improve that metric, obtaining the results available in Figure 4 and a R-square of 10.02%. Although it is an improvement, the unrealistic values (e.g.: one predicted in the order of the 600 billion days) and low R-square value deems this classifier as unreliable of yet, needing improvement.

IV. Conclusions and Future Work

With the aim of improving the automatic processing of Github issues, we use a Natural Language analysis approach

in order to provide developers with a tool able to fulfill the tasks of:

- Detection of version and issue cross-referencing.
- Suggestion of related labels.
- Listing similar issues.
- Prediction of the time needed to close the issue.

According to the obtained results, the developed system is able to identify the version and issue cross-references, as well able to list similar issues based on the issue text.

For the suggestion of related labels, an One vs Rest Classifier over a Logistic Regression of the data is able to provide reliable suggestions for some of the considered labels, needing an increase on the training dataset for the less reliable labels. As for the prediction of the duration, the produced model has obtained terrible results and is considered unusable at its current state. On future works, in order for this tool to be improved, the following suggestions should be taken into account:

- Better computation method for the issue duration, since it is based on the issue dates and not on the commit history related to it, that would better reflect the time used to work on the issue.
- Filtering of issues claimed as duplicate, due to early closing.
- Utilize temporal data mining analysis for a more accurate and reliable duration prediction.

Though this work used the Terasology repository as its subject, the methodology followed can be applied to other repositories for their analysis.

REFERENCES

[1] C. Solis and X. Wang, "A study of the characteristics of behaviour driven development," in *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, Aug 2011, pp. 383–387.

[2] C. Shearer, "The CRISP-DM model: The new blueprint for data mining," *Journal of Data Warehousing*, 2000.

[3] 2019 GitHub, Inc. Github. [Online]. Available: https://github.com/

[4] MovingBlocks. Terasology. [Online]. Available: https://github.com/MovingBlocks/Terasology

[5] 2019 GitHub, Inc.

[6] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ser. ETMTNLP '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 63–70. [Online]. Available: https://doi.org/10.3115/1118108.1118117

PAPERS IN ALPHABETICAL ORDER

AUTHORS IN ALPHABETICAL ORDER

Amir Hossein Farzamiyan
André Coelho
David Freitas
Ehsan Shahri
Fernec Tam'asi
Flávio Couto
Hajar Baghcheband
Juliana Paula Felix, Enio Vasconcelos Filho and Flávio Henrique Teles Vieira
Leonardo Ferreira
Luis Roque
Mafalda Falcão Ferreira, Rui Camacho and Luís Filipe Teixeira
Miguel Abreu
Paula Silva
Pedro Peixoto


9 789727 522439 >

U.PORTO

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

DEI DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA

DSIE´19
14th Doctoral Symposium
in Informatics Engineering

Faculty of Engineering
University of Porto
Porto | Portugal