



ARTSS: Perspektīvās tehnoloģijas noturīgiem un drošiem servisiem

VPP-COVID-2020/1-0009

Transporta līdzekļa ierašanās laika prognozēšanas (TILP) algoritma implementācija

Kontaktpersona: Jānis Grabis, grabis@rtu.lv

Autori: Jānis Kampars, Jānis Grabis, Gunti Mosāns

Versija, datums: 1. versija, 28.12.2020.

1 Pamatojums

Transportlīdzeklis pārvietojas noteiktā maršrutā atbilstoši grafikam. Tā pieturvietas un paredzētais ierašanās laiks ir zināmas. Šī informācija ir publiski pieejama atvērto datu veidā Latvijas atvērto datu portālā (<https://data.gov.lv/dati/lv/dataset/atd-gtfs>). Transportlīdzekļa atrašanās vietu fiksē kases aparātā integrētā GPS ierīce un šie dati ir pieejami transporta pārvadājumu uzņēmuma autoparka vadības sistēmā. GPS datiem ir jāveic priekšapstrāde, jo jēldatos:

1. Netiek identificētas apmeklētās un nākamās pieturvietas, un transportlīdzekļa apstāšanās var notikt dažādu iemeslu dēļ;
2. Netiek norādīta maršruta reisa informācija, un iegūtos GPS datus ir nepieciešamas piesaistīt noteiktam maršruta reisam.

Maršruta īstenošanas laikā dažādu ārējo faktoru ietekmes rezultātā (piem., pasažieru plūsma, ceļu segumu kvalitāte, satiksmes negadījumi) ir iespējamas nobīdes no plānotā maršruta izpildes grafika. Ja nobīdes ir zināmas savlaicīgi, tad ir iespējams brīdināt pasažierus vai veikt citus nepieciešamos pielāgojumus. Ierašanās laika prognozēšana, ņemot vērā pašreizējos datus par grafika izpildi un vēsturisko novērojumus, ļauj savlaicīgi identificēt sagaidāmās nobīdes.

Prognozēšanas algoritms aktīvajiem maršrutiem (pašlaik kustībā esošajiem transportlīdzekļiem) nosaka transportlīdzekļa ierašanās laiku nākamajās pieturvietās. Algoritma implementācija nodrošina pilnu datu apstrādes ciklu no GPS jēldatu iegūšanas līdz prognožu iegūšanai, iesakot prognozēšanas modeļa apmācīšanu.

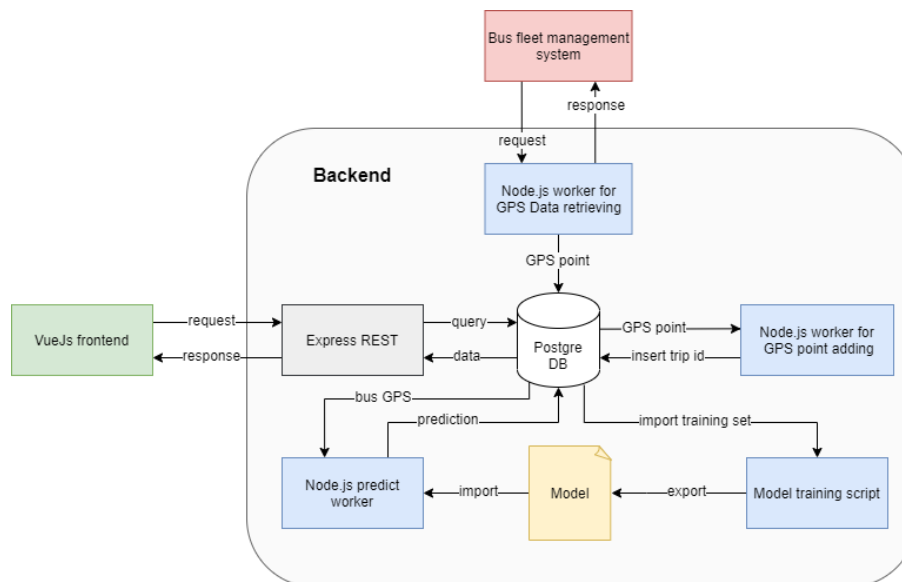
2 Prognozēšanas algoritms

Lai noteiktu prognozējamo transportlīdzekļa ierašanās laiku pieturvietā, ir izstrādāts TILP prognozēšanas algoritms. Algoritmu veido septiņas komponentes (1. attēls):

1. “Node.js predict worker” – NodeJS un Python implementēts modulis transportlīdzekļa ierašanās laika prognozēšanai nākamajās pieturās atbilstoši identificētajam maršruta reisam;
2. “Node.js worker for GPS data retrieving” – GPS jēldatu ielasīšana no autoparka vadības sistēmas;
3. “Node.js worker for GPS point adding” – nosaka, kuram maršruta reisam pieder saņemtais GPS datu punkts;
4. “Express REST” – tīmekļa serviss, kas paredzēts maršrutu telpisko datu un prognožu iegūšanai;
5. “Model training script - nodrošina priekšapstrādāto GPS datu pārveidošanu Python modeļa apmācības vajadzībām nepieciešamajā formātā;
6. “Model” – Python scikit-learn implementēts prognozēšanas modelis (atsevišķs modelis katram maršrutam);
7. “Postgre DB” – PostgreSQL relāciju datubāze ar PostGIS paplašinājumu GPS punktu un maršrutu informācijas uzglabāšanai.

“Bus fleet management system” ir ārējā komponente, transporta pārvadājumu uzņēmuma autoparka vadības sistēma, no kuras tiek izgūta informācija par transportlīdzekļa atrašanās vietu.

“Vue.js frontend” ir priekšgala komponente, kas attēlo prognozēšanas rezultātus un tiek pielāgota atbilstoši klienta vajadzībām.



1. attēls. TILP algoritma komponentes.

Ierašanās laika prognozes iegūst “Node.js predict worker”. Prognožu iegūšanai izmanto ieejas datus:

- autobusa reisu;
- iepriekšējo apmeklēto pieturu;
- nākamo pieturu;
- veikto distanci no iepriekšējās pieturas;
- atlikušo distanci līdz nākamajai pieturai;
- diennakts laiku;
- nedēļas dienu;
- paredzēto ierašanās laiku atbilstoši grafikam.

Šī informācija tiek nodota prognozēšanas modelim, kas atgriež prognozēto ierašanās laiku nākamajā pieturā.

3 TILP implementācija

TILP ir implementēts atbilstoši strādņu lietojumprogrammatūras arhitektūrai. TILP implementācijai ir izmantota JavaScript programmēšanas valodas un tās Node.js un Express ietvari un SQL strukturētā vaicājumvaloda. TILP implementācijas atkarības ir:

1. Node.js 15.2.1 (MIT licence)
2. Express.js 4.17.1 (MIT licence)
3. PostgreDB 12.1 (PostgreSQL licence atvasināta no MT licences)
4. PostgreSQL PostGIS paplašinājums (Creative Commons Attribution-Share Alike)
5. Scikit-learn (BSD licence)

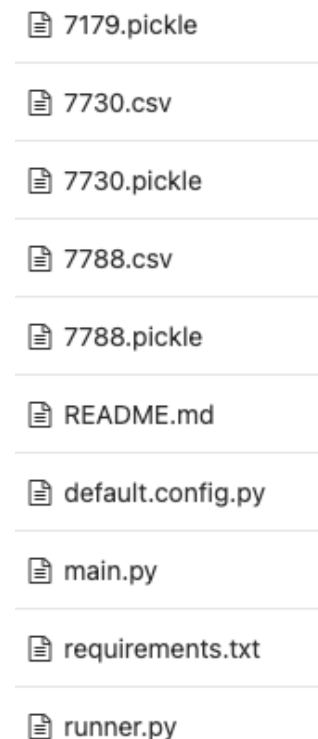
6. VueJS 2.6.11(MIT licence)

TILP pirmkods glabājas GitLab repozitorijā <https://gitlab.vitk.lv/TILP/tlp-backend>.

3.1 Koda struktūra

Aizmugursistēmas python komponente satur mašīnmācības modeļu apmācības un izpildes skriptus, kas implementēta Python programmēšanas valodā un balstās uz scikit-learn bibliotēku. Šīs komponentes koda struktūra attēlota 2. attēlā un to veido:

- maršrutu mašīnmācības modeļu apmācības dati csv formātā (piemēram, 7730.csv);
- maršrutu mašīnmācības modeļu binārie faili (piemēram, 7179.pickle);
- lietošanas instrukcijas (README.md);
- konfigurācijas faila paraugs (default.config.py);
- mašīnmācības modeļu apmācības skripts (main.py);
- python virtuālās vides atkarību fails (requirements.txt);
- mašīnmācības modeļa izpildes fails (runner.py).



2. attēls. Python aizmugursistēmas komponente

Aizmugursistēmas NodeJS komponente nodrošina tīmekļa servisu, gps datu apstrādi, prognožu iegūšanu, mijiedarbojoties ar Python komponenti. NodeJS komponentes galvenās sastāvdaļas ir:

- tīmekļa serviss prognožu un telpisko datu iegūšanai, integrācijai ar priekšgalsistēmu (bin/www, app.js, routes/index.js);

- konfigurācijas faila piemērs (default.config.js);
- datubāzes shēma (db_schema.sql);
- algoritma konstantes (constants.js);
- datubāzes piekļuves bibliotēka (db.js);
- GPS datu priekšapstrādes skripts (gps.js);
- ATD atvērto datu priekšapstrādes skripts (graph_builder.js);
- NodeJS bibliotēku atkarību fails (package.json);
- Prognožu iegūšanas skripts (predict.js);
- Mašīnmācības modeļu apmācības kopas sagatavošanas skripts (prepare_ml_data.js);
- GPS datu apstrādes, prognožu iegūšanas, mašīnmācības modeļu apmācības kopu sagatavošanas skriptu palaišanas skripts (runner.js);
- Aizmugursistēmas risinājuma palaišanas skripts (startup_script.sh).

bin	Initial commit
public/stylesheets	Initial commit
python	Remove unneeded files
routes	Initial commit
views	Initial commit
.gitignore	Initial commit
README.md	Additional instructions
app.js	Initial commit
constants.js	Initial commit
db.js	Initial commit
db_schema.sql	Initial commit
default.config.js	Initial commit
gps.js	Initial commit
graph_builder.js	Initial commit
package-lock.json	Initial commit
package.json	Initial commit
predict.js	Initial commit
prepare_ml_data.js	Initial commit
runner.js	Initial commit
startup_script.sh	Initial commit

3. attēls. NodeJS aizmugursistēmas komponente

Priekšgalsstēmas VueJS komponente nodrošina aizmugursistēmas REST servisu izsaukumus, datu apstrādi un datu grafisko attēlošanu lietotāja interfeisā. VueJS komponentes galvenās sastāvdaļas ir:

- VueJS lietotnes izstrādes un produkcijas vides konfigurācijas fails (vue.config.js);
- lietotnes bibliotēku un komandu saraksts (package.json);
- VueJS lietotnes inicializācijas fails (src/main.js);
- VueJS lietotnes pamata karkass (src/App.vue);
- VueJS lietotnes maršrutu (saišu) definēšanas fails (src/router.js);
- lokālās krātuves datu glabāšanas fails (src/store.js);
- transporta prognozēšanas lietotnes grafiskā datu attēlošanas un REST servisu apstrādes komponente (src/views/dashboard/Routes.vue);
- kodola VueJS komponentu katalogs (src/views/dashboard/components/core);
- pielāgotu komponentu katalogs (src/views/dashboard/component);
- katalogs ar Sass failiem, kas nodrošina lietotnes dizainu (src/sass);
- katalogs ar lokalizācijas failiem, kurus izmanto teksta tulkošanai (src/locales);
- lietotnes pamata veidnes komponentes (src/components/base);
- dizaina attēlu katalogs (src/assets);
- publisko failu katalogs (public);
- lietotnes pamata izpildīšanas fails (public/index.html);
- katalogs ar testiem (tests);

public	Push to gitlab
src	Push to gitlab
tests	Push to gitlab
.browserslistrc	Push to gitlab
.editorconfig	Push to gitlab
.env	Push to gitlab
.eslintrc.js	Push to gitlab
.gitignore	Push to gitlab
CHANGELOG.md	Push to gitlab
ISSUE_TEMPLATE.md	Push to gitlab
LICENSE.md	Push to gitlab
README.md	CHANGE README
babel.config.js	Push to gitlab
cypress.json	Push to gitlab
jest.config.js	Push to gitlab
now.json	Push to gitlab
package.json	Push to gitlab
postcss.config.js	Push to gitlab
vue.config.js	Push to gitlab

4. attēls VueJS priekšgalsistēmas komponente

3.2 Ieejas dati

ATD nodrošina datus (tajā skaitā telpiskos datus) par maršrutiem, reisiem, pieturvietām un grafikam atbilstošajiem ierašanās laikiem.

Autoparka vadības sistēma nodrošina autobusa kustības datus, kas satur GPS koordinātes, koordināšu fiksēšanas laiku un atbilstošo maršrutu.

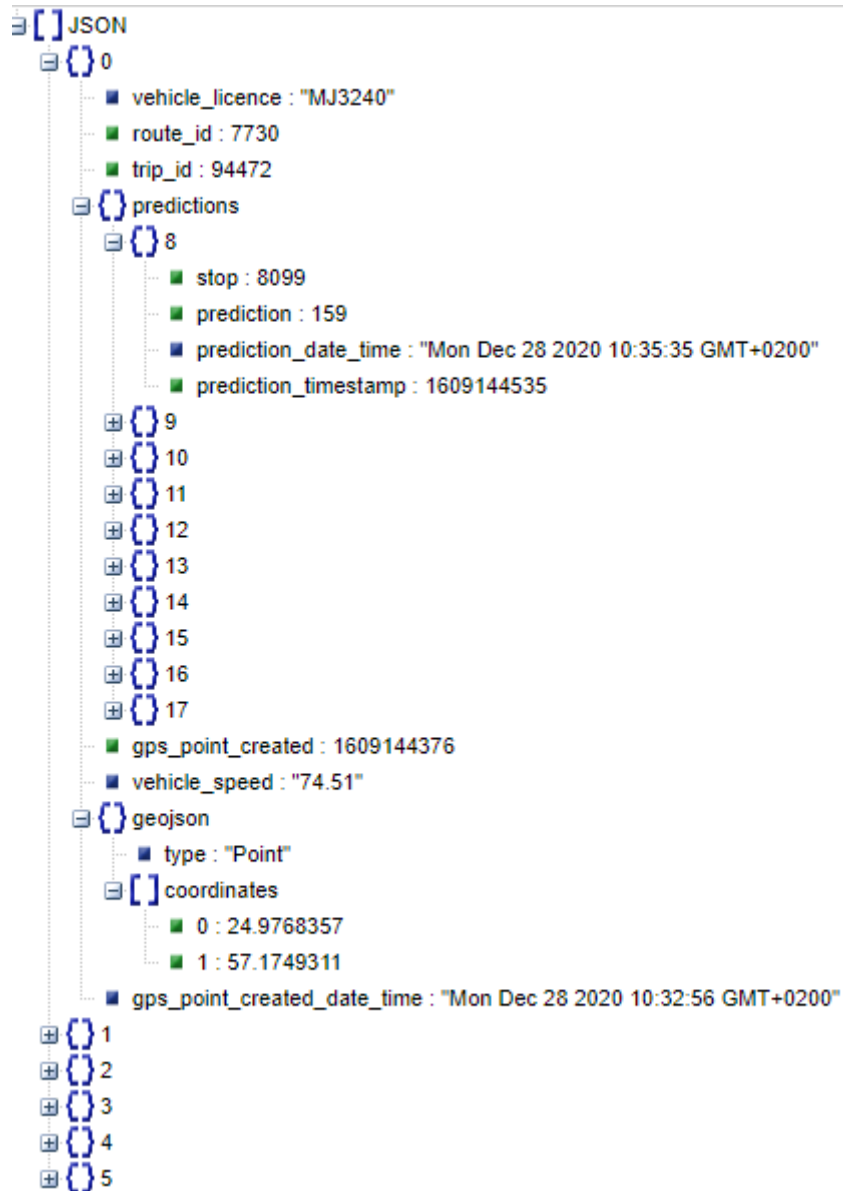
3.3 TILP API

Aktuālās transportlīdzekļa ierašanās laika prognozes iegūts, izpildot:

GET <http://atd.vitk.lv:5000/predictions>

Servisa atbildes piemērs JSON formātā ir dots 1. pielikumā. Atbildes grafisks attēlojums ir dots 4. attēlā. Serviss atgriež visu aktīvo maršrutu masīvu. Katram maršrutam tiek parādīta:

1. Informācija par transportlīdzekli un maršrutu;
2. Prognozes visām nākamajām pieturvietām;
3. Transportlīdzekļa atrašanās vieta.



4. Attēls. TILP API atbildes JSON dokumenta piemērs

4 Pielikums

```
[
  {
    "vehicle_licence": "MJ3240",
    "route_id": 7730,
    "trip_id": 94472,
    "predictions": {
      "8": {
        "stop": 8099,
        "prediction": 159,
        "prediction_date_time": "Mon Dec 28 2020 10:35:35 GMT+0200",
        "prediction_timestamp": 1609144535
      },
      "9": {
```



```
    "stop": 8096,
    "prediction": 185,
    "prediction_date_time": "Mon Dec 28 2020 10:38:40 GMT+0200",
    "prediction_timestamp": 1609144720
  },
  "10": {
    "stop": 8094,
    "prediction": 238,
    "prediction_date_time": "Mon Dec 28 2020 10:42:38 GMT+0200",
    "prediction_timestamp": 1609144958
  },
  "11": {
    "stop": 8090,
    "prediction": 419,
    "prediction_date_time": "Mon Dec 28 2020 10:49:37 GMT+0200",
    "prediction_timestamp": 1609145377
  },
  "12": {
    "stop": 8088,
    "prediction": 242,
    "prediction_date_time": "Mon Dec 28 2020 10:53:39 GMT+0200",
    "prediction_timestamp": 1609145619
  },
  "13": {
    "stop": 11761,
    "prediction": 235,
    "prediction_date_time": "Mon Dec 28 2020 10:57:34 GMT+0200",
    "prediction_timestamp": 1609145854
  },
  "14": {
    "stop": 8080,
    "prediction": 363,
    "prediction_date_time": "Mon Dec 28 2020 11:03:37 GMT+0200",
    "prediction_timestamp": 1609146217
  },
  "15": {
    "stop": 8076,
    "prediction": 476,
    "prediction_date_time": "Mon Dec 28 2020 11:11:33 GMT+0200",
    "prediction_timestamp": 1609146693
  },
  "16": {
    "stop": 15758,
    "prediction": 1107,
    "prediction_date_time": "Mon Dec 28 2020 11:30:00 GMT+0200",
    "prediction_timestamp": 1609147800
  },
  "17": {
    "stop": 11382,
    "prediction": 1207,
    "prediction_date_time": "Mon Dec 28 2020 11:50:07 GMT+0200",
    "prediction_timestamp": 1609149007
  }
},
"gps_point_created": 1609144376,
"vehicle_speed": "74.51",
"geojson": {
  "type": "Point",
  "coordinates": [
    24.9768357,
    57.1749311
  ]
},
"gps_point_created_date_time": "Mon Dec 28 2020 10:32:56 GMT+0200"
}
]
```