

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Tarkvarateaduse instituut

Henri Rütmaa 191579IABB

**ANDMETE VISUALISEERIMINE ASJADE INTERNETI VÕGUS**

Bakalaureusetöö

Juhendaja: Inna Švartsman  
Magistrikraad

Tallinn 2019

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Henri Rüütmaa

20.05.2019

## **Annotatsioon**

Lõputöö eesmärk on andmete visualiseerimine IoT-võrgu jaoks telekommunikatsiooniettevõtte näitel. Lõputöös luuakse visualiseerimislahendus, mis pakub ettevõttele võimaluse analüüsida oma IoT võrku ja võrgus olevate seadmete informatsiooni.

Töö eesmärgi saavutamiseks leitakse sobivate kriteeriumite alusel võimalikud andmeanalüüsiplatvormid, mida omavahel võrreldakse. Kui sobiv vahend on leitud, luuakse automaatselt uuenevad visuaalid, millega monitooritakse võrgu ning seadmete andmeid. Andmeanalüüsiplatvormiga visualiseerimine peab tooma praktilist kasu jooniste näol, mille järgi on võimalik tuvastada puudusi võrgu töös ning muuta võrgu toimimist efektiivsemaks. Loodud lahenduse kaudu on ettevõttel võimalik andmeid lihtsamalt analüüsida : jälgida kaardil tugijaamade ning seadmete asukohti; tuvastada mustreid tugijaamate koormatuse kohta ajas; kontrollida, kas andmed saabuvad järjest ning millistel seadmetel võib leviga probleeme olla.

Lõputöö on kirjutatud eesti keeles ning sisaldab 33 lehekülge teksti, sh 5 peatükki, 34 joonist ja 2 tabelit.

## **Abstract**

### **Data visualization for the IoT network**

The aim of the thesis is to visualize the data for the IoT network on the example of a telecommunications company. The achieved visualization solution provides comprehensive opportunity for the company to analyze the information of IoT network.

In order to achieve the goal of the work, possible data analysis platforms that are compared with each other are found on the basis of suitable criteria. Once a suitable tool is found, automatically updated visuals are created to monitor network and device data. Visualization with the selected data analysis platform must result in practical benefits in terms of visual drawings that can identify network problems and make the network more efficient. Through the solution, the company can more easily analyze data - track the locations of base stations and equipment on the map, identify different patterns of base station load over time, check whether data arrives sequentially and which devices may have problems with the signal.

The thesis is in Estonian and contains 33 pages of text, 5 chapters, 34 figures, 2 tables.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , Rakendustarkvaraliides,
Backup	Andmete varundamine
Dashboard	Visualisatsioonide kogum
dBm	Detsibellides mõõdetud võimsuse suhe ühe millivati kohta
Docer	Tarkvarapakettide ja platvorm-teenusepakkumiste kogum
ELK stack	Lühend kogumist Elasticsearch, Logstash ja Kibana
FSK	Digitaalsete andmete diskreetne andmeedastus
gRPC	Google'i loodud kaugprotseduuri suhtlus
IoT ehk Internet of things	Asjade internet
Index	Andmekogum
JSON	<i>JavaScript Object Notation</i> , andmeedastusvorming
LoRa	Traadita andmete saatmise tehnoloogia
LPWAN	Vähese energiatarbe ja laia ulatusega traadita sidevõrk
Lucene	Tasuta ja avatud lähtekoodiga infootsingutarkvara kogum
Mapping	Kaardistatakse saabuva dokumendi indekseerimine
Mount	Programmile asukoha ettenäitamine
NoSQL	Baasiks oleva andmemudeli <i>SQL</i> puudumine
Template	Mall indekseerimiseks
RESTful	Tarkvaraarhitektuuri stiil veebiteenuste arendamisel
RSSI	<i>Received signal strength indicator</i> , signaalitugevuse näitaja
SNR	<i>Signal to Noise Ratio</i> , signaali ja müra suhe
SQL	<i>Structured Query Language</i> , andmebaasi päringukeel

# Sisukord

Jooniste loetelu .....	7
Tabelite loetelu .....	8
1 Sissejuhatus .....	9
2 Teoreetiline ülevaade .....	10
2.1 Visualiseerimine .....	10
2.2 IoT võrgud .....	10
2.3 LoRa võrk .....	10
2.4 LoRa-server .....	11
3 Andmeanalüüsiplatvormi valik ja visualiseerimine valitud platvormiga.....	12
3.1 Andmeanalüüsiplatvormi valik.....	12
3.1.1 Nõuded platvormile .....	12
3.1.2 Platvormide võrdlus .....	13
3.2 Valitud platvormi kirjeldus.....	14
3.2.1 Elasticsearch.....	14
3.2.2 Kibana.....	14
3.2.3 Logstash.....	15
3.3 Andmete kogumine Logstashiga .....	15
3.4 Saabuvate andmestruktuuride kuju.....	19
3.5 Visualiseerimisel kasutatavate andmeväljade kirjeldused.....	21
3.6 Mapping template' i modifitseerimine .....	22
3.7 Indeksite loomine Kibanas .....	23
3.8 Skriptitud väljad .....	24
3.9 Visualisatsioonid .....	26
3.9.1 Visualiseerimisega alustamine .....	26
3.9.2 Visualisatsioonide loomine .....	27
3.9.3 Juhtpaneel loomine.....	36
4 Tulemused .....	38
5 Kokkuvõte .....	40
Kasutatud kirjandus .....	41
Lisa 1 Mappingu template näide .....	43

## Jooniste loetelu

Joonis 1. Status pipeline .....	16
Joonis 2. Stats pipeline .....	17
Joonis 3. Pipelines.yml-faili sisu .....	17
Joonis 4. Käivitud Logstashi pipeline'ed.....	18
Joonis 5. Logstashi failide mount .....	19
Joonis 6. Logstashi käivituskäsk .....	19
Joonis 7. Asukoha automaatne kuju .....	23
Joonis 8. Asukoha muudetud kuju .....	23
Joonis 9. Kibana Indexi loomine .....	24
Joonis 10. Vajaliku time-välja valimine .....	24
Joonis 11. Skriptitud väli seadmetüübi nime jaoks .....	25
Joonis 12. Skriptitud väli viimati aktiivse tugijaama jaoks .....	25
Joonis 13. Skriptitud väli ühendatud tugijaamade nimede jaoks .....	25
Joonis 14. Skriptitud väli järjenumbriga vea tuvastamiseks .....	25
Joonis 15. Skriptitud väli andmeliikuvuse jaoks .....	25
Joonis 16. Skriptitud väli leidmaks muude seadmete pakettide arvu.....	25
Joonis 17. Kibana pealeht.....	26
Joonis 18. Kibana indeksi väljad .....	26
Joonis 19. Kibana visualisatsioonitüübid.....	27
Joonis 20. Tugijaamade Coordinate Map.....	28
Joonis 21. Tugijaamade Heat Map .....	28
Joonis 22. LoRa võrgukoormuse Vertical Bar ühe tunni täpsusega .....	29
Joonis 23. Väljade kasutamine Data Table visualisatsioonil .....	30
Joonis 24. Data Table LoRa-tugijaama info .....	31
Joonis 25. Seadmete Coordinate Map .....	32
Joonis 26. Seadmete Data Table .....	32
Joonis 27. Seadmete SNR Area joonis.....	33
Joonis 28. Seadmete RSSI Line joonis.....	34
Joonis 29. Seadmete tüüp Pie Chart .....	35
Joonis 30. Pakettide järjenumbriga Vertical Bar .....	36
Joonis 31. Salvestatud otsingu kuvamine .....	36
Joonis 32. Juhtpaneelide loomine Kibanas .....	37
Joonis 33. Tugijaamade juhtpaneel.....	38
Joonis 34. Seadmete juhtpaneel .....	39

## Tabelite loetelu

Tabel 1. Platvormide võrdlus .....	13
Tabel 2. Saabuvate andmestruktuuride kuju .....	20



# 1 Sissejuhatus

Lõputöö teema on “Andmete visualiseerimine asjade interneti võrgus” ning see Bakalaureusetöö kirjutatakse ühe Eesti telekommunikatsiooniettevõtte näitel.

Vaadeldavas telekommunikatsiooniettevõttes on asjade interneti (ingl *Internet of things* ehk IoT) võrk alles loomisel ning kõiki andmeid ei kasutata piisavalt ära.

Lõputöö eesmärgiks on luua olemasolevate andmete põhjal visualisatsioonid. Seega on vaja kõnealused andmed kokku koguda ja visualiseerida, et ennetada/tuvastada võrgus esinevaid probleeme ja muuta võrgu kasutus efektiivsemaks.

Eesmärgi saavutamiseks tuleb täita järgmised ülesanded:

- kaardistada ettevõtte nõuded;
- leida nõuetekohane platvorm;
- seadistada valitud platvorm;
- visualiseerida andmed.

Töö käigus leitakse sobivate kriteeriumite alusel võimalikud andmeanalüüsiplatvormid, mida omavahel võrreldakse. Kui sobiv vahend on leitud, luuakse automaatselt uuenevad visuaalidest koosnevad juhtpaneelid, millega võrgu ning seadmete andmeid monitooritakse. Andmeanalüüsiplatvormiga visualiseerimine tulemuseks peab tooma praktilist kasu jooniste näol, mille järgi on võimalik tuvastada puudusi võrgu töös ning muuta võrgu toimimine efektiivsemaks. Loodud lahenduse kaudu peab olema ettevõttel võimalik andmeid lihtsamalt analüüsida : jälgida tugijaamade ning seadmete asukohti kaardil; tuvastada mustreid tugijaamade koormatuse kohta ajas; kontrollida, kas andmed saabuvad järjest ning millistel seadmetel võib leviga probleeme olla.

## 2 Teoreetiline ülevaade

Peatükis antakse ülevaade kasutatavast IoT võrgust ning kirjeldatakse LoRa võrgu ja serveri põhimõtteid.

### 2.1 Visualiseerimine

Mitmete eri valdkondade praktikud ja akadeemikud esitavad andmeid graafiliselt. Geograafid, majandusteadlased, sõjaväelased, statistikud, insenerid, bioloogid ning paljud muud spetsialistid peavad andmeid mõistma graafiliselt. [1]

Psühholoogia uurib andmete tajumist ning seda, kuidas me tajume mõne elemendi mõju, nt värvid ja kujundid. Arvutiteadus ja statistika on töötanud välja mitmeid uusi valdkondi nagu masinõpe ja andmete kaevandamise meetodid. Graafilised ja multimeediakujundused on kriitilise tähtsusega infopaneelide ehitamisel ning see avaldub ka dünaamiliste infopaneelide loomisel. Infopaneelid realiseeruvad mitmes elemendis: andmed, kaalud, jooned, tulbad või värvitud kujud. Andmete visualiseerimisel on erinevaid trende: infograafia üldistus, liikuvus, sotsiaalmeedia, geolokaliseerimine, suurandmed, globaliseerumine ja teaduse suhtlemine. [1]

### 2.2 IoT võrgud

Asjade interneti väga kiire arengu tõttu on populaarseks osutunud LPWAN-võrgutehnoloogia. Eeldatavalt on aastaks 2020 selles võrgus ligikaudu 50 miljardit traadita seadet. LPWAN on suurt ala kattev võrk, mille sees kasutatakse võimalikult väikeseid andmemahte, et minimeerida andmeid saatva seadme energiakasutust. Madal energiakasutus teeb võimalikuks selle, et patareisid on vaja vahetada väga harva.[2]

See lõputöö on orienteeritud andmetele, mida kasutatakse LoRa võrgus.

### 2.3 LoRa võrk

LoRa-võrgu eelis on võimekus katta suur ala. Üks tugijaam võib katta kogu linna või sadu ruutkilomeetreid. signaali levi kvaliteet sõltub paljuski asustustihedusest. LoRa võrgus

olevad seadmed ei ole seotud konkreetse tugijaamaga, nt üks seade võib saata andmed korraga mitmesse tugijaama. Iga tugijaam, mis on internetiga ühendatud, saadab andmed edasi serverisse, see annab omakorda võimaluse muuta LoRa-seade mobiilseks.[3]

Seadmed töötavad LoRa-võrgus asünkroonselt ja suhtlevad ainult siis, kui neil on andmed edastamiseks valmis, andmed võivad tekkida sündmusepõhiselt või nende edastamine võib olla planeeritud mingi ajavahemiku järel.[3]

Võimaldamaks suure ulatusega võrku, peab tugijaamal olema võimekus võtta vastu väga suurtes kogustes andmeid. LoRa-võrgus saavutatakse see kohandades andmete edastuskiirus tning kasutades tugijaamas mitmesageduslikku vastuvõttu . Seega on võimalik võtta vastu erinevaid sõnumeid eri kanalitelt. Võrgukoormust mõjutab see, mitut kanalit seade korraga kasutab, mis on andmeedastuskiirus ja andmemaht ning kui tihti on seadmel vaja andmeid edastada.[3]

LoRa kasutab Euroopas kümmet kanalit, kaheksa nendest on andmekiirustele 250bps kuni 5,5 kbps, üksik andmeedastuskiirus LoRa-kanalil on 11kbps ning üksikul FSK-kanalil on 50 kbps. Euroopa maksimaalne väljundvõimsus on +14 dBm erandina on *G3-band*'is lubatud +27 dBm.[3]

## **2.4 LoRa-server**

LoRa-server on avatud lähtekoodiga rakenduste kogum. LoRa server võtab vastu andmed, mis seadmed on tugijaamedele saatnud. LoRa server on lüli enne seda, kui andmed jõuavad rakendustesse. Server pakub tugijaamade, rakenduste, samuti rakendustega seotud seadmete haldamiseks eri võimalusi.[5]

LoRa-server on disainitud nii, et seda saaks väga paindlikult kasutada, näiteks rakendab LoRa Appi serveri komponent rakendusserveri komponenti ja pakub läbi kasutajaliidese võimaluse hallata tugijaamasid, rakendusi ning seadmeid. Süsteemi saab kasutada ka programmirakenduse kaudu, mida rakendatakse gRPC ja JSON REST API-des. Lisaks on API-d konstrueeritud selliselt, et allsüsteeme võib asendada muu tarkvaraga, mis kasutab samu liideseid. [5]

## **3 Andmeanalüüsiplatvormi valik ja visualiseerimine valitud platvormiga**

### **3.1 Andmeanalüüsiplatvormi valik**

Selles peatükis kirjeldatakse, mis nõuetele peab vastama andmeanalüüsi platvorm ning valitakse välja nendest sobivaim. Kui sobilik andmeanalüüsi platvorm on valitud, siis seadistatakse see ning visualiseeritakse andmed.

#### **3.1.1 Nõuded platvormile**

Peatükis kirjeldatakse nõudeid, mis andmeanalüüsi platvormile esitatakse. Nõuetes on lähtunud olemasolevatest andmetest ning toetunud ettevõtte vajadustele.

- Süsteem peab olema tasuta
- Süsteem peab olema avatud lähtekoodiga.
- Süsteem peab töötama Windowsi operatsioonisüsteemis.
- Süsteem peab töötama Linuxi operatsioonisüsteemis.
- Süsteemis peab olema võimalik viia kokku erinevate allikate andmed.
- Süsteemis peab olema võimalik luua graafikuid ja tabeleid.
- Süsteemis peab olema võimalik väärtustega graafikuid filtreerida.
- Süsteem peab kuvama kaardil seamete/tugijaamade asukohta .
- Süsteem peab kuvama kaardil tugijaamade koormatust.
- Süsteem peab kuvama kaardil seadmete signaalitugevusi.
- Süsteem peab kuvama eraldi iga tugijaama koormatust mingil ajavahemikul.
- Süsteem peab kuvama mitu korda on seade saatnud mingi ajavahemiku vahel tugijaamale infot

- Süsteem peab kuvama, millal seade viimati tugijaamale andmeid saatis.
- Süsteem peab kuvama RSSI/SNR-graafiku, mis on seotud konkreetse seadme ja tugijaamaga.
- Süsteem peab kuvama graafiku, kus oleks näha, kui suur on ühe või teise seadmetüübi osakaal andmeedastuses.
- Süsteem peab kuvama visualisatsiooni, mis võimaldaks kontrollida, kas pakettide järjenumbrid tulevad järjestikku.

### 3.1.2 Platvormide võrdlus

Tabelis 1 on nimetatud võimalikud alternatiivid andmeanalüüsiplatvormi valimisel, samuti on kirjeldatud valimise jaoks olulised kriteeriumid. Andmeanalüüsi platvormi valikul loodi nõuete põhjal kriteeriumid: Kas platvormi kasutamine on tasuta? Kas platvorm on avatud lähtekoodiga? Kas platvorm omab kaardirakenduse olemasolu? Kas platvormi on võimalik kasutada Linuxi operatsioonisüsteemil? Kas platvorm omab enda andmebaasi? Tabelis võrreldakse omavahel Elastic Stacki, Grafanat, MS Power BI'd ja ThingsBoard'i.

Tabel 1. Platvormide võrdlus

	Tasuta	Avatud lähtekood	Kaardirakenduse olemasolu	Linux	Oma andmebaas
Elastic Stack	+	+	+	+	+
Grafana	+	-	+	+	-
MS Power BI	-	-	+	-	-
ThingsBoard	-	+	+	+	-

Visualiseerimisplatvormiks valiti Kibana, mis kuulub Elastic Stacki rakenduste alla. Kibana kasuks rääkis avatud lähtekood ning võimalus seda oma serveri olemasolul kasutada tasuta. Kibanal on ka tasuline variant, aga eelpool toodud nõuete tõttu puudub vajadus seda kasutada.

Kuna programmi on vajalik käivitada Linuxi serveris, on selle toetus olemas. Andmed salvestatakse platvormi enda Elasticsearchi NoSQL-i andmebaasi.[4]

## 3.2 Valitud platvormi kirjeldus

### 3.2.1 Elasticsearch

Elasticsearch on avatud lähtekoodiga NoSQL-i andmebaas. See baseerib otsingumootoril Lucene, mis on ehitatud RESTful API-de jaoks. Platvormi saab lihtsalt kasutusele võtta, see on maksimaalselt usaldusväärne ja lihtsalt hallatav. Elasticsearch pakub ka täiustatud päringuid detailanalüüsi tegemiseks ja säilitab kõik andmed tsentraalselt. Viimane on kasulik dokumentide kiire otsingu teostamiseks.[4]

Elasticsearch võimaldab salvestada, otsida ja analüüsida suuri andmemahutusi. Enamasti volitatakse sellega otsingumootoritele vastavaid rakendusi, ent seda saab kasutada ka kaasaegsete veebi- ja mobiilirakenduste otsingumootori platvormides. Lisaks kiirotsingule pakub tööriist ka keerulisi analüüse ja paljusid täiustatud funktsioone.[4]

Infot kogutakse indeksitesse, mille ajaline pikkus on tavaliselt üks kuu, aga võib olla ka näiteks üks päev või üks aasta. Serverist saadetakse IoT-võrgus olevad andmed JSON'i kujul. Andmed edastatakse hetkel, kui need on seadmelt serverisse saabunud, iga saabumine kirjeldatase sündmusena ehk eraldi event'ina, mis salvestatakse Elasticsearchi indeksisse. [4]

### 3.2.2 Kibana

Kibana on kasutajaliides, mis võimaldab otsida, visualiseerida ja analüüsida, kasutades Elasticsearchis olevaid andmeid.[4]

Kibana on andmete visualiseerimise tööriist, mis täiustab ELK stacki. Tööriista kasutatakse Elasticsearchi dokumentides olevate andmete visualiseerimiseks ning see aitab arendajal teha kiire sissevaate andmetesse. Kibana *dashboard* ehk juhtpaneel pakub mitmesuguseid interaktiivseid diagramme, georuumilisi andmeid ja annab võimaluse neid koos esitada. Kibana saab kasutada Elasticsearchi kataloogidesse talletatud andmete otsimiseks, vaatamiseks ja nendele päringute tegemiseks. Kibana aitab läbi viia täiustatud andmeanalüüsi ja visualiseerida oma andmeid mitmesugustes tabelites, diagrammides ja kaartidel. [4]

### 3.2.3 Logstash

Logstash on andmekorje tööriist, mis kogub andmesisendeid ning sisestab need Elasticsearchi. Logstash kogub eri tüüpi andmed, mis pärinevad eriandmeallikatest ja muudab need hiljem kasutatavaks. Logstash suudab ühendada eri allikate andmeid ning need soovitud sihtkohtadesse viia. See võimaldab kõik andmed analüütika jaoks puhastada, paigutada ja visualiseerida. Logstash koosneb kolmest komponendist:

- Input – andmete läbimine, konverteerimaks nende formaat arvutile arusaadavaks.
- Filter – teatud toimingute või sündmuste kogum, kus defineeritakse muudatuste tingimused.
- Output – Kirjeldatakse kuhu esitatakse andmeid On võimalik sisestada erinevaid andmeid näiteks – json, csv või log faile. [4]

### 3.3 Andmete kogumine Logstashiga

Andmete korjamiseks kasutati Logstashi kahte käsku, millega vajalikud andmed edastati Elasticsearchi. *Input*'is kirjeldatakse, kus andmed paiknevad ning millisel ajavahemikul neid korjatakse. Oluline on ka defineerida, millises formaadis loetavad andmed on. Andmed uuenevad minuti tagant ja on *JSON*-i formaadis. Kuna andmed on juba õiges formaadis, jääb järgmine filter tühjaks. *Output*'is kirjeldatakse, milline peab olema loodava Elasticsearchi indeksi nimi ning kui tihti peab uue indeksi tekitama. Loodava konfiguratsiooni näitel asuvad need Linux serveris kaustas:

```
„/usr/share/logstash/pipeline/“[6]
```

Loodud Logstashi käsud juhendite järgi:[7]

Joonisel 1 on JSON-tüüpi andmete korjamiseks Logstahi käsk. Andmed võetakse olemasolevalt http aadressilt ning edastatakse localhostis olevale elasticseach'i pordile. Luuakse indeks status pikkusega üks kuu.

```

input {
  http_poller {
    urls => {
      test2 => {
        method => get
        url => "andmete asukohaks http link"
      }
    }
    request_timeout => 60
    schedule => { cron => "* * * * * UTC" }
    codec => "json"
  }
}

filter{}

output {
  elasticsearch {
    hosts => "elasticsearch:9200"
    index => "status_%{+MM_YYYY}"
  }
}

```

Joonis 1. Status pipeline

Joonisel 2 on JSON-tüüpi andmete korjamiseks Logstahi käsk. Andmed võetakse olemasolevalt http aadressilt ning edastatakse localhostis olevale elasticsearch'i pordile. Luuakse indeks stats pikkusega üks kuu.



```

input {
  http_poller {
    urls => {
      test2 => {
        method => get
        url => " andmete asukohaks http link "
      }
    }
    request_timeout => 60
    schedule => { cron => "* * * * * UTC" }
    codec => "json"
  }
}

filter{}

output {
  elasticsearch {
    hosts => "elasticsearch:9200"
    index => "stats_%{+MM_YYYY}"
  }
}

```

Joonis 2. Stats pipeline

Üldjuhul on Logstashis main pipelines, millega on võimalik töösse määrata korraga üks konfiguratsioon, mitme korraga jooksva konfiguratsiooni puhul tuleb need defineerida Logstashis pipelines.yml-failis. Joonisel 3 on näidatud selle faili sisu.[8]

```



- pipeline.id: status
  path.config: "/usr/share/logstash/pipeline/status.conf"
- pipeline.id: stats
  path.config: "/usr/share/logstash/pipeline/stats.conf"

```

Joonis 3. Pipelines.yml-faili sisu

Joonisel 4 on kuvatud käivitunud pipeline'd.

Nodes: 1 Memory: 302.4 MB / 990.8 MB Events Received: 309.5k Events Emitted: 309.5k

ID ↑	Events Emitted Rate
stats	 0 e/s
status	 0 e/s

Joonis 4. Käivitunud Logstashi pipeline'ed

Elasticsearch, Kibana ja Logstash on Linuxis serveris Docker'i konteinerites, seega tuleb ka failid ka mount'ida, et käivitusel oskaks programm vajalikke faile leida ja käivitada. Joonisel 5 on kuvatud selle faili sisu. [9]

```
logstash:
  extends:
    file: ../../common.yml
    service: logstash_common
  volumes:
    - ./logstash-config:/config-dir
    - ./logstash-config/pipeline/:/usr/share/logstash/pipeline/
    - ./logstash-config/logstash-
conf/pipelines.yml:/usr/share/logstash/config/pipelines.yml
  command: logstash
```

Joonis 5. Logstashi failide mount

Kui konfiguratsioonid ning mount'id on tehtud, tuleb Logstash käivitada Docker'is käsuga:

```
sudo docker start logstash [10]
```

Joonis 6. Logstashi käivituskäsk

### 3.4 Saabuvate andmestruktuuride kuju

Tabelis 2 on esitatud Logstashist ja LoRa-serverist saabuvate andmete mall *JSON* kujul. Sellised nimi-väärtus paarid saavad Elasticsearchi baasi, kust on hiljem võimalik visualiseerida need andmed Kibanas.

Tabel 2. Saabuvate andmestruktuuride kuju

Stats.json	Online.json	rxInfo.json
<pre>[ { mac: " ", name: "", timestamp: "", start: "", end: "", rxPacketsReceived:, rxPacketsReceivedOK:, txPacketsReceived:, txPacketsEmitted: } ]</pre>	<pre>[ { mac: "", name: "", description:" ", location: { lat:, lon: }, altitude:, createdAt: "", updatedAt: "", firstSeenAt: "", lastSeenAt: "", organizationID: "", channelConfigurationID: "", ping:, gwRxPacketsReceived: }, ]</pre>	<pre>{applicationID: "", "applicationName": "", "nodeName": "", "devEUI": "", "rxInfo": [{ "mac": "", "rssi": , "loRaSNR": , "name": "", "latitude":, "longitude": "altitude": 0, "location": { "lat": , "lon":}}] "txInfo": { "frequency":, "dataRate": { "modulation": "", "bandwidth": , "spreadFactor": }, "adr":, "codeRate": "" }, "fCnt": , "fPort": , "data": { "decoded": { "data": { "value":, "metaData": [ ] } }, "raw": "", "reading": { "multiplier":, "unit": "", "value": }}, "sensorId": 1, "consumable": true, location{}}},</pre>

### 3.5 Visualiseerimisel kasutatavate andmeväljade kirjeldused

Stats.json ja Online.json andmed puudutavad LoRa-tugijaama infot: need uuenevad minuti jooksul. rxInfo uueneb siis, kui seadmelt saabub ühele või enamale tugijaamale uut infot. Andmete rxInfo.json edastamiseks on ettevõtte poolt ühendus LoRa serveri ning Elasticsearchi baasi vahel. Stats.json ja Online.json jaoks on ettevõtte poolt loodud eraldi http lehtedena väljundid mis uuenevad iga minut tagant.

#### **Stats**

mac - tugijaama MAC aadress

name - tugijaama nimi

timestamp – hetkeae

start - mõõdistamisvahemiku algus

end - mõõdistamisvahemiku lõpp

rxPacketsReceived - ajavahemiku jooksul saabunud pakettide arv

rxPacketsReceivedOK - ajavahemiku jooksul saabunud meie andmepakettide arv

txPacketsReceived – saabunud tagasisidepakettide arv

txPacketsEmitted –vastatud tagasisidepakettide arv

#### **Online**

mac - tugijaama MAC aadress

name - tugijaama nimi

description – tugijaama kirjelduse lisamine

location - tugijaama asukoha koordinaadid

lat – laiuskraad

lon – pikkuskraad

altitude – kõrgus

lastSeenAt – viimati saabunud info kellaeg

gwRxPacketsReceived - ajavahemiku jooskul saabunud meie andmepakkettide arv

## **rxInfo**

nodeName –tugijaamale infot saatnud seadme nimi

devEUI – seadme unikaalne kood

location – seadme asukoht

rxInfo – ühe või enama tugijaama ühenduse info

rxInfo.mac – tugijaama MAC aadress

rxInfo.rssi - loodud signaali tugevus dBm

rxInfo.loRaSNR - signaali ja müra suhe

rxInfo.name - tugijaama nimi

rxinfo.location – tugijaama asukoht: pikkus- ja laiuskraad

fCnt – saadetud paketi järjenumbr

Need on olemasolevad andmed mis salvestatakse Elasticsearchi indeksitesse.

### **3.6 Mapping template' i modifitseerimine**

Elasticsearch saab üldjuhul andmestike põhjal aru, mis on JSON'i formaadis, kuid teatud tingimustes antakse Elasticsearchile ette mapping andmete õige kuju jaoks. Kui sama struktuuriga andmed tulevad korduvalt, näiteks iga kuu, siis on vaja luua tekkivatele indeksitele template, st template'i järgi tehakse igale indeksile iga kuu jaoks automaatselt mapping, mille järgi on võimalik omakorda teha indeksi andmekuju Kibana jaoks. Template'i muutmine toimub käskude PUT, GET ja DELETE abil. [11]

Muudetud template'i on vaja selles töös käsitletud näite puhul näiteks kaardirakenduse jaoks. Kaardirakenduse koordinaatide jaoks on vaja muuta andmekuju. [12]

Kogu mapping template on toodud lisas 1.

Joonisel 7 on kujutatud Elasticserarch'i automaatselt genereeritud *mapping*:

```
"location": {
  "properties": {
    "lat": {
      "type": "float"
    },
    "lon": {
      "type": "float"
    }
  }
}
```

Joonis 7 Asukoha automaatne kuju

Joonisel 8 on template'is etteantud uus mapping:

```
"location": {
  "type": "geo_point"
}
```

Joonis 8. Asukoha muudetud kuju

### 3.7 Indeksite loomine Kibanas

Visualisatsioonide loomiseks on vaja defineerida Kibana indeks, millel peab olema sama nimi, kui Elasticsearchi indeksil. Näiteks kui Elasticsearchi indeksi nimetus on events\_01\_2019 (näitab, et indeksi ajaline pikkus on üks kuu), peab Kibana indeks olema events\*. Ainult sel juhul on Kibana indeksil ligipääs kõikidele samanimelistele Elasticsearchi indeksitele sõltuvalt kuust või aastaarvust. [13]

Joonisel 9 on kuvatud Kibana indeksi loomist.

## Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

**Step 1 of 2: Define index pattern**

Index pattern

You can use a \* as a wildcard in your index pattern.  
You can't use spaces or the characters \, /, ?, ", <, >, |.

✓ **Success!** Your index pattern matches 10 indices.

---

events\_01\_2019

Joonis 9. Kibana Indexi loomine

Sõltuvalt saabuvatest andmetest, on vaja valida, millist ajavälja indeksil kasutatakse, sõltuvalt sellest järjestatakse kõik andmed, mis juba Elasticsearchi on indeksis või alles saabuavad. [13]

Joonisel 10 on kuvatud *time*-välja valmist Kibana indeksile

## Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.

**Step 2 of 2: Configure settings**

You've defined `stat*` as your index pattern. Now you can specify some settings before we create it.

Time Filter field name Refresh

@timestamp  
createdAt  
end  
firstSeenAt  
lastSeenAt  
start  
timestamp  
updatedAt  
\_\_\_\_\_

I don't want to use the Time Filter

Joonis 10. Vajaliku time-välja valimine

## 3.8 Skriptitud väljad

Kibanas on võimalik väikeste programmijupikeste abil luua skriptitud väljasid, mille abil on võimalik visualiseerimist lihtsustada. Kibanas on ka võimalik ebavajalikke andmeid olemasolevate väljade nime järgi peita. Kibana ei tee Elasticsearchi andmetes muudatusi.[14] Visuaalide jaoks loodud väljad:

- Device type - juhul kui seadme/seadmegrupi nimi ei ole üheselt määratav, saab sellele kasutusmugavuse tõstmiseks anda käsitsi ette loogilise nimetuse.



```
if ( params._source.nodeName == "xmeter000"){ return "Electricity  
Meter"}
```

Joonis 11. Skriptitud väli seadmetüübi nime jaoks

- Last Active - tugijaamade puhul on vaja teada, millal need viimati aktiivsed olid.

```
new Date().getTime() - doc['rxInfo.time'].value.getMillis()  
new Date().getTime() - doc['lastSeenAt'].value.getMillis()
```

Joonis 12. Skriptitud väli viimati aktiivse tugijaama jaoks

- Connected gateways - tugijaamad, millega ühendus loodi

```
doc['rxInfo.name.keyword'].values
```

Joonis 13. Skriptitud väli ühendatud tugijaamade nimede jaoks

- Check Fcnt – kontrollib pakettide järjenumbrit, kuvab teate, kui väli on 0

```
if ( doc['fCnt'].value == 0){ return "fCnt error"} else { return  
"fCnt is ok" }
```

Joonis 14. Skriptitud väli järjenumbriga vea tuvastamiseks

- SUM – kontrollib kogu võrgu andmeliikuvuse mahtu

```
doc['rxPacketsReceived'].value + doc['txPacketsReceived'].value +  
doc['txPacketsEmitted'].value
```

Joonis 15. Skriptitud väli andmeliikuvuse jaoks

- Delta – leiab andmeid, mis ei tule ettevõtte registreeritud seadmetest.

```
doc['rxPacketsReceived'].value - doc['rxPacketsReceivedOK'].value
```

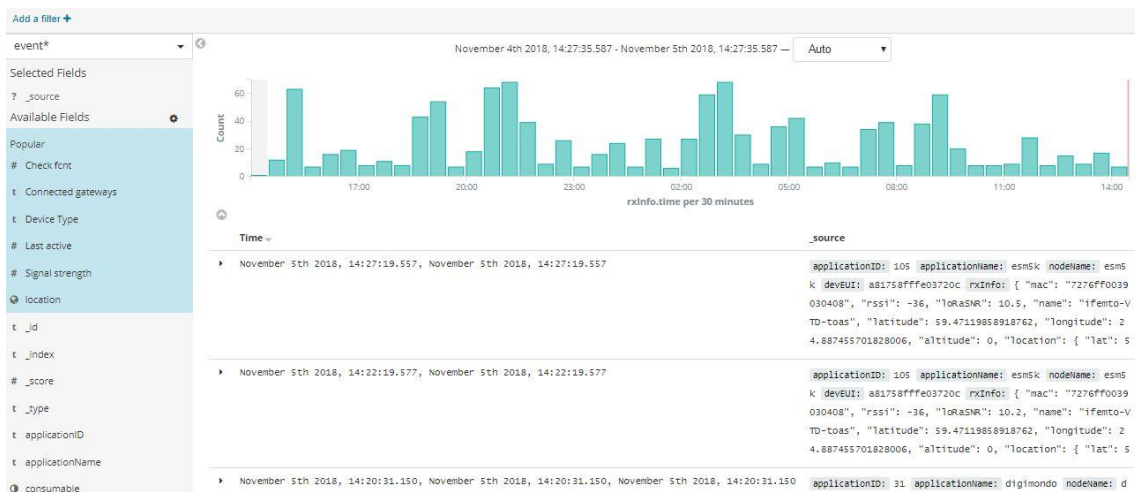
Joonis 16. Skriptitud väli leidmaks muude seadmete pakettide arvu

## 3.9 Visualisatsioonid

### 3.9.1 Visualiseerimisega alustamine

Kibana pealehel saab Kibana indeksi nimetuse järgi vaadata kõiki saabunud andmeid, neid on võimalik filtreerida aja või konkreetse välja otsinguparameetri järgi. Võimalik on näha otsingupäringule vastavate dokumentide arvu ja saada statistikat andmete kohta. Kui time-väli on valitud indeksil nii konfigureeritud, siis kuvatakse nende andmete kohta histogramm. [15]

Joonisel 17 on kuvatud Kibana pealeht, kus on näha eelnevalt mainitud informatsiooni.



Joonis 17. Kibana pealeht

Joonis 18 kujutab Kibana indeksi välja ning nende tüüpe. Samuti on võimalik valida vasakult millist indeksit soovitakse vaadelda.

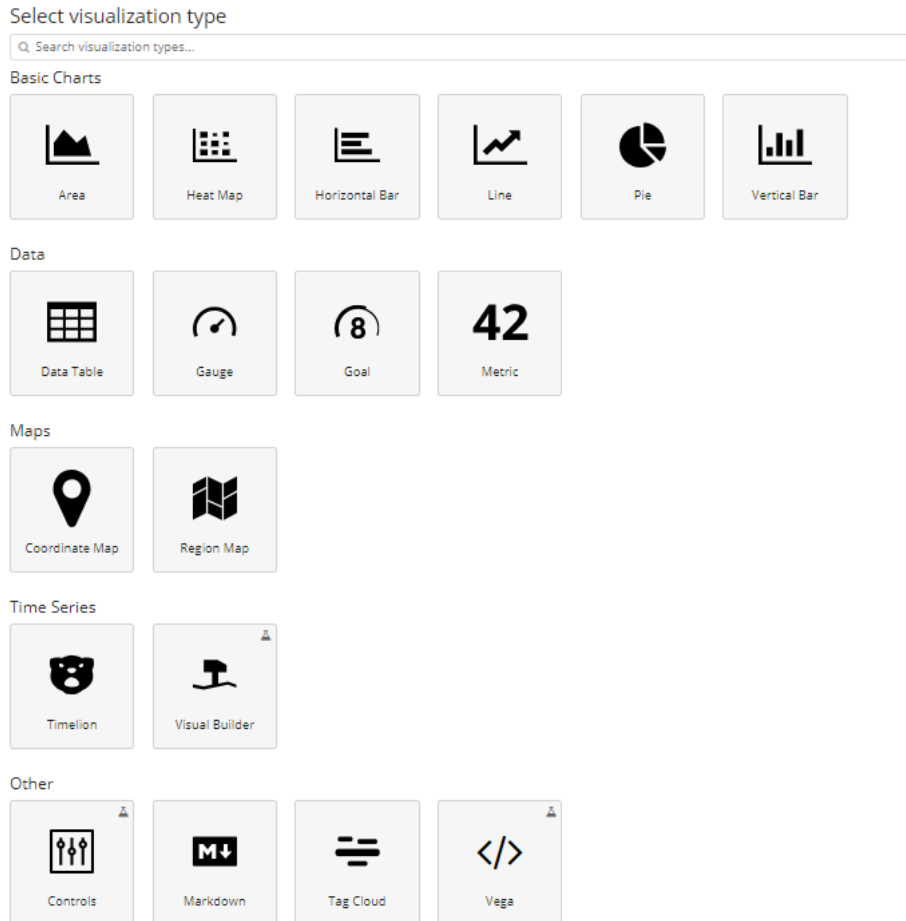
The screenshot shows the Kibana 'Index Patterns' page for the 'event\*' index. It displays a table of fields with their types, formats, and searchability. The table includes fields like '\_id', '\_index', '\_score', '\_source', '\_type', 'applicationID', 'applicationID.keyword', and 'applicationName'.

Name	Type	Format	Searchable	Aggregatable	Excluded
_id	string		●	●	
_index	string		●	●	
_score	number				
_source	_source				
_type	string		●	●	
applicationID	string		●		
applicationID.keyword	string		●	●	
applicationName	string		●		

Joonis 18. Kibana indeksi väljad

Alustamaks visualiseerimisega tuleb valida tüüp, mille alusel andmeid visualiseerida. [16]

Joonis 19 kujutab kõnealuse versiooni kõiki võimalikke tüüpe.

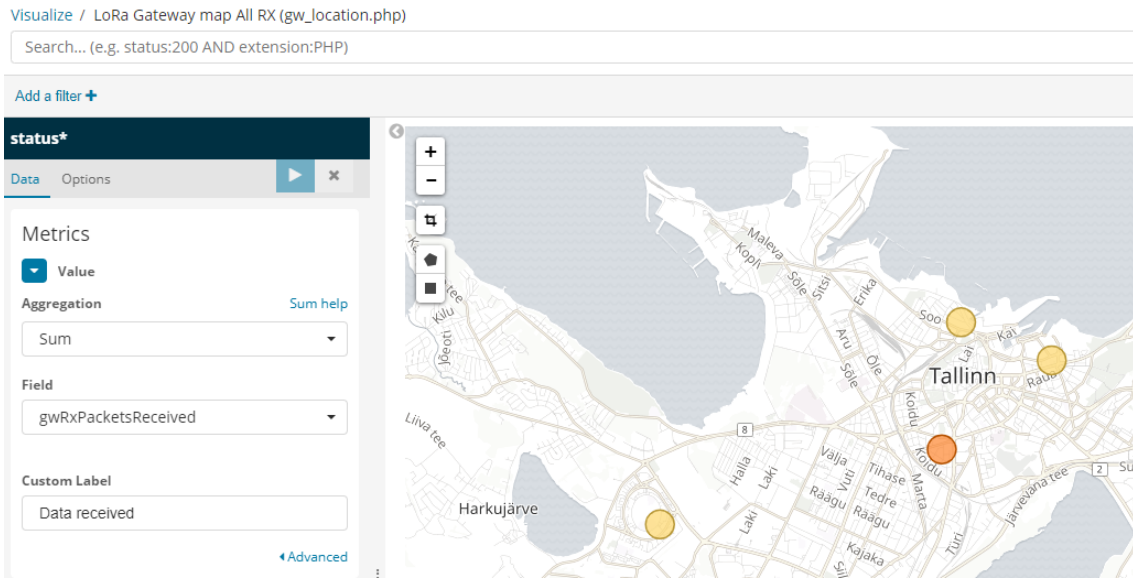


Joonis 19. Kibana visualisatsioonitüübid

### 3.9.2 Visualisatsioonide loomine

Esimene visuaal on koordinaatidega kaardi kuvamine valikus „Coordinate Map“. Selleks kasutatakse välja `rxInfo.location`, mis omab tugijaama koordinaate. Kaardil kuvatakse tugijaamu, mille koormatus on esitatud väljal `gwRxPacketsReceived` Viimati nimetatud väli värvib koordinaadi saabunud pakettide arvu alusel helekollasest – punaseni vastavalt saabunud pakettide arvule. Mida rohkem pakette saadeti, seda punasem on kaardil tugijaam. Värviskaala täpsem seletus on kaardi nurgas legendil. Kui kaardil hiirega liikuda, on võimalik näha ka konkreetse tugijaama koormatust, samuti selle koordinaate.[17]

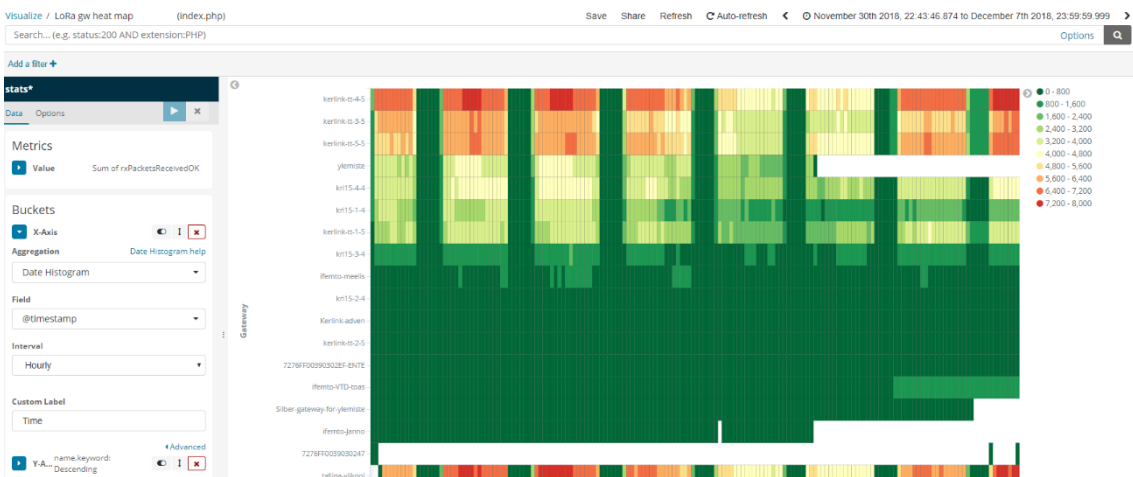
Joonisel 20 on kuvatud tugijaamade kaart.



Joonis 20. Tugijaamade Coordinate Map

Järgmine visualisatsioon kasutab tüüpi Heat Map, millele on lisatud mitu välja. Väli name näitab tugijaama nimetust, timestamp andmete saabumise aega, andmete, väli rxPacketsReceivedOK. Visualisatsiooni kaudu on võimalik leida mustreid konkreetse tugijaama koormatuse kohta ajas. Eri värvid võimaldavad leida koormatuimad tugijaamad.[18]

Joonisel 21 on kuvatud tugijaamade Heat Map



Joonis 21. Tugijaamade Heat Map

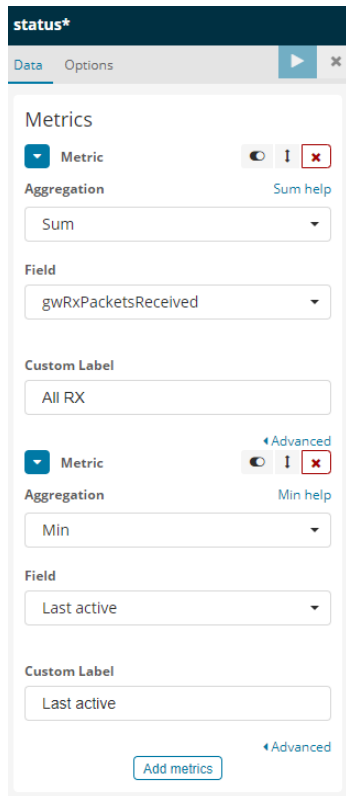
Kolmandal visualisatsioonil on kuvatud Vertical Bar.[19] See näitab graafikut pakettide arvu kohta. Lisatud on väljad rxPacketsReceivedOK, Delta, txPacketsEmitted ja txPacketsReceived, kõik nimetatud väljad on summeeritud sõltuvalt valitud ajast.

Joonisel 22 on välja timestamp kaudu valitud üheks pulgaks üks tund. Graafiku abil on võimalik kuvada ettevõttele neile kuuluvate seadmete saadetisi muudest LoRa-seadmetest.



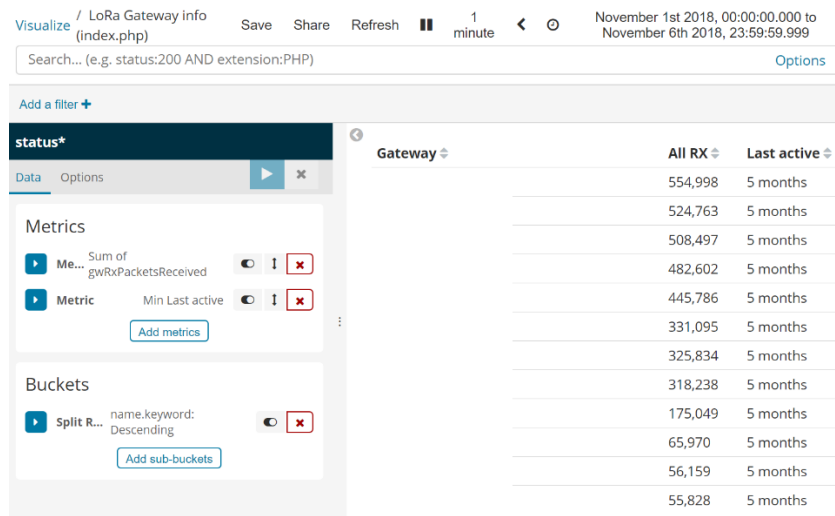
Joonis 22. LoRa võrgukoormuse Vertical Bar ühe tunni täpsusega

Neljandal (vt Joonis 23) visuaalil on kujutatud eraldi iga tugijaama koormatust mingi ajavahemiku jooksul. Samuti kuvatakse, millal tugijaam viimati võrgus oli. Visualisatsiooni tüübiks on Data Table. Juhul kui filtrit ei muudeta, siis on see vaikimisi viimased 24 tundi. [20]



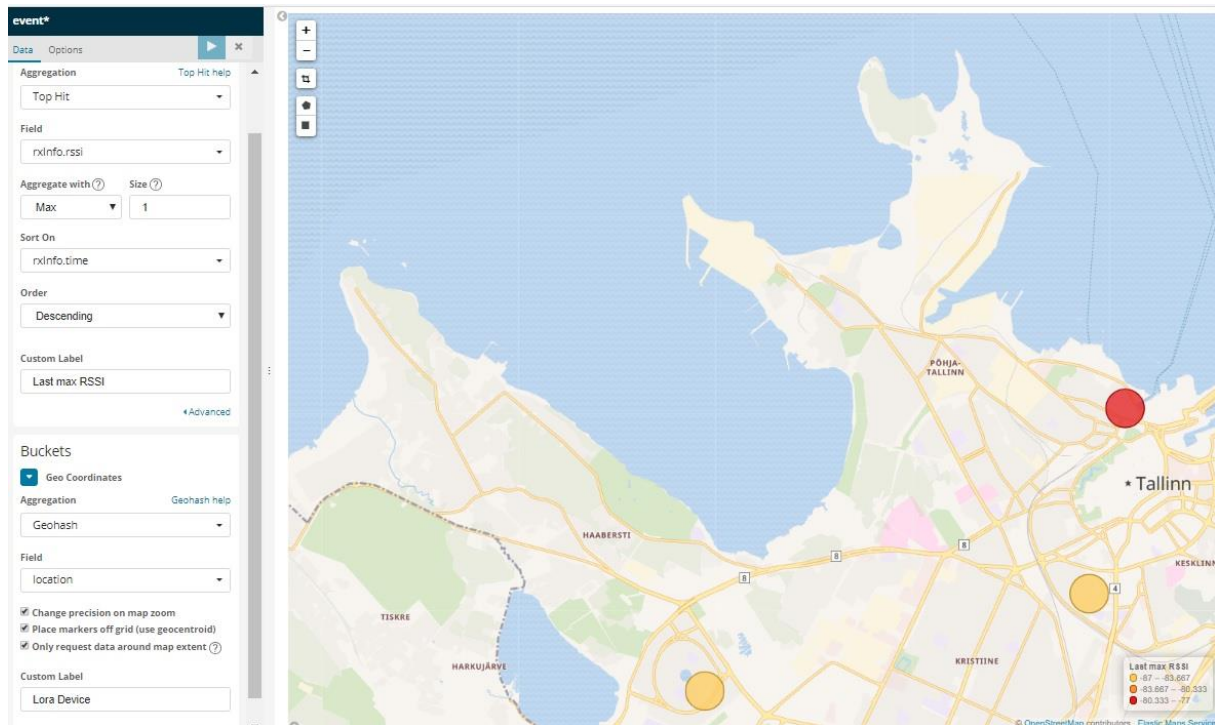
Joonis 23. Väljade kasutamine Data Table visualisatsioonis

Joonisel 24 on tabel, mis kuvab konkreetsete tugijaamadele saabuvate pakettide arvu ning seda millal viimane pakett saabus.



Joonis 24. Data Table LoRa-tugijaama info

Järgmine visuaal (vt Joonis 25) on kaart LoRa seadmete jaoks. Kasutatakse välja location, millel on seadmete koordinaadid. Signaalitugevus saadakse väljalt rxInfo.rssi Kaardil kuvatakse seadmete asukoht, värvid helekollasest – punaseni märgivad signaalitugevust, kus helekollane on kõige nõrgem signaal ja punane kõige tugevam signaal. Kaardi nurgas legendil on skaala, mis värvikoodi avab ja võimaldab otsustada, mis vahemikku jääb viimase sessiooni signaalitugevus. Kui kaardil hiirega liikuda, on võimalik näha ka konkreetse seadme viimase sessiooni signaalitugevust, samuti selle seadme koordinaate. [17]



Joonis 25. Seadmete Coordinate Map

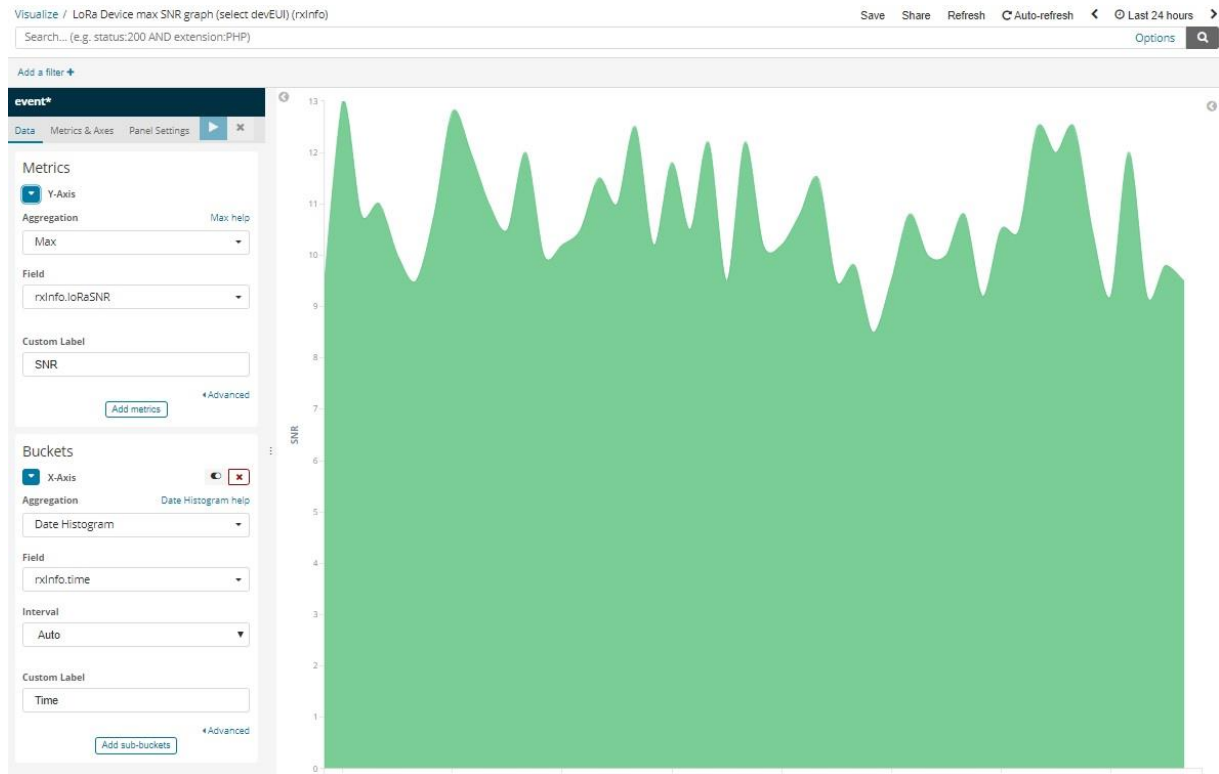
Järgmine visuaal on Data Table (vt Joonis 26), mis kuvab tabelina seadme unikaalseid nimetusi; devEUI, mis kuvab seadmetüüpe ning tugijaamu, millega seade ühenduse lõi ning loeb kokku mitu korda see valitud aja jooksul tugijaamale pakette saatis, kuvatakse ka viimse ühenduse signaalitugevus ning selle aeg. [20]



Joonis 26. Seadmete Data Table

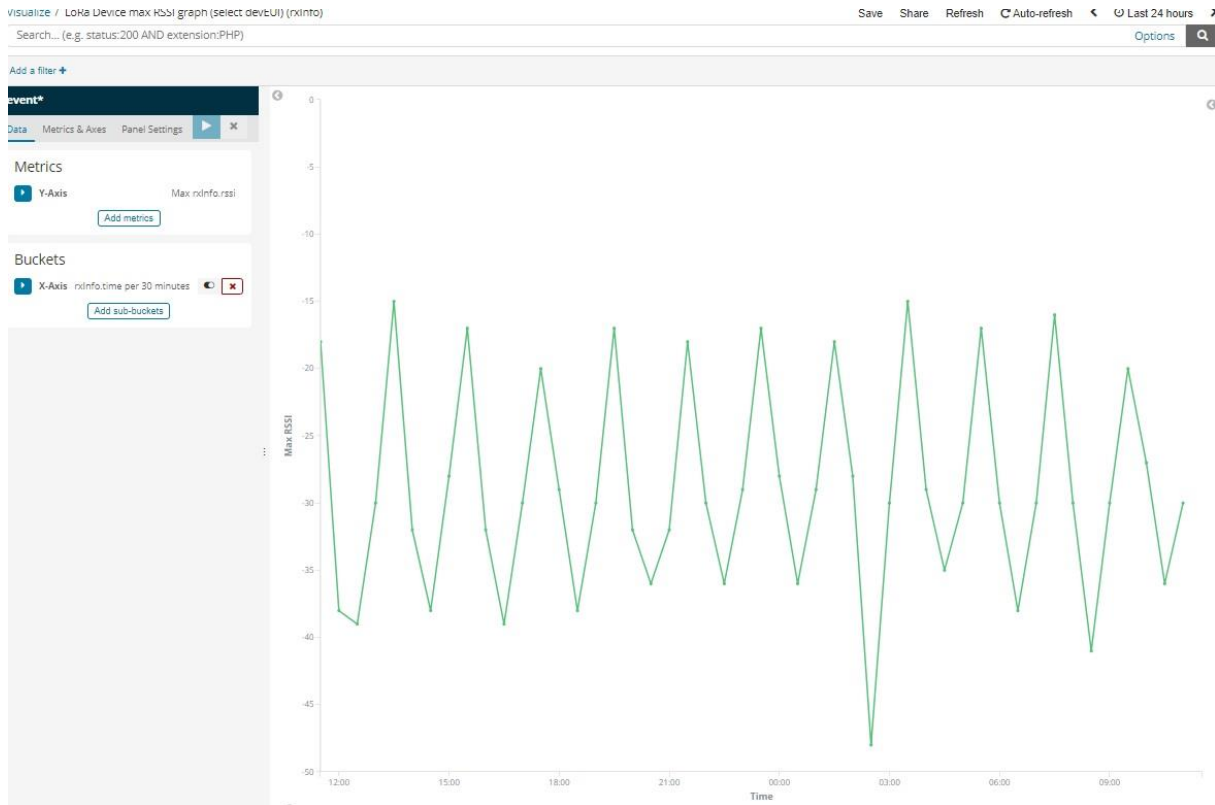


Visuaal Area (vt Joonis 27) kuvab signaali ja müra suhet ajas konkreetse tugijaama ja seadme vahelise ühenduse analüüsimiseks. Kasutatakse andmeid väljadelt rxInfo.loRaSNR ja rxInfo.time. [19]



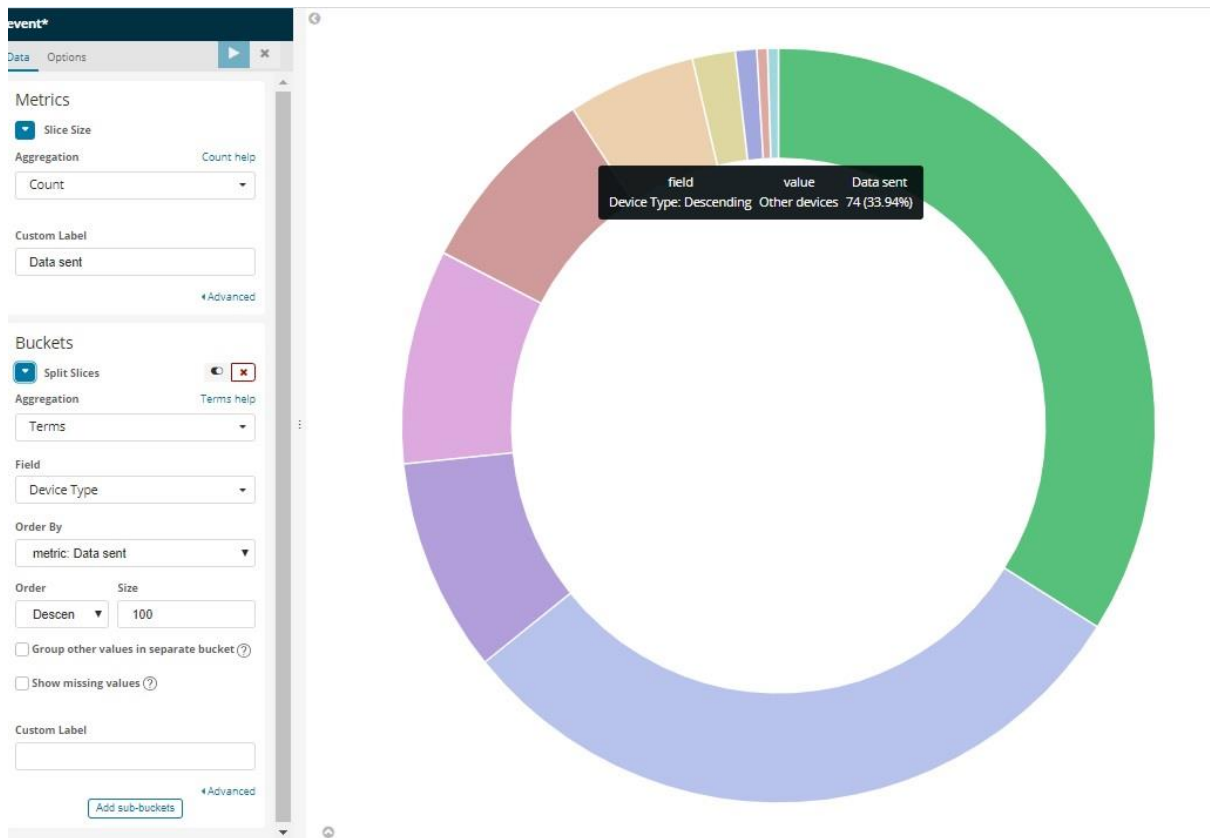
Joonis 27. Seadmete SNR Area joonis

Visuaal Line (vt Joonis 28) kuvab signaalitugevust ajas mingi konkreetse tugijaama ja seadme vahelise ühenduse analüüsimiseks. Kuvatakse maksimaalset signaalitugevust ühendatud tugijaamade nimekirjast. [19]



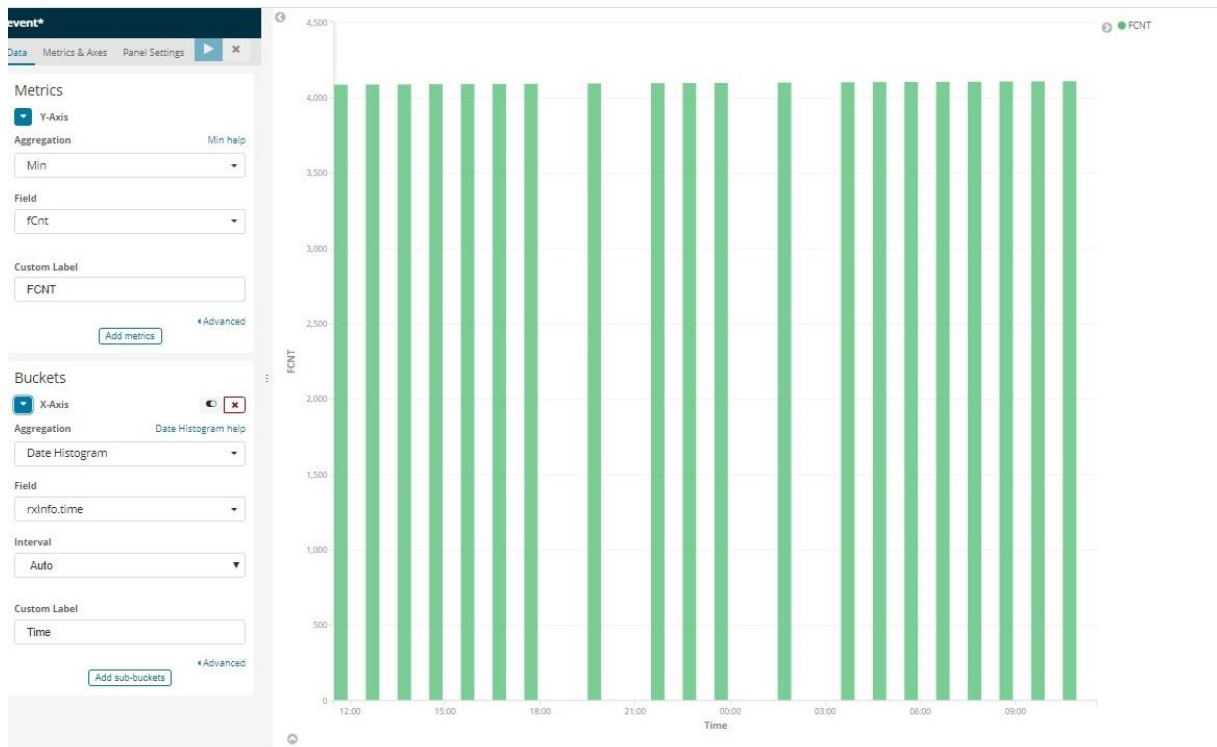
Joonis 28. Seadmete RSSI Line joonis

Järgmine visuaal Pie Chart (vt Joonis 29) võimaldab kasutajal näha erinevate seadmetüüpide saadetiste osakaalu võrgus. [21]



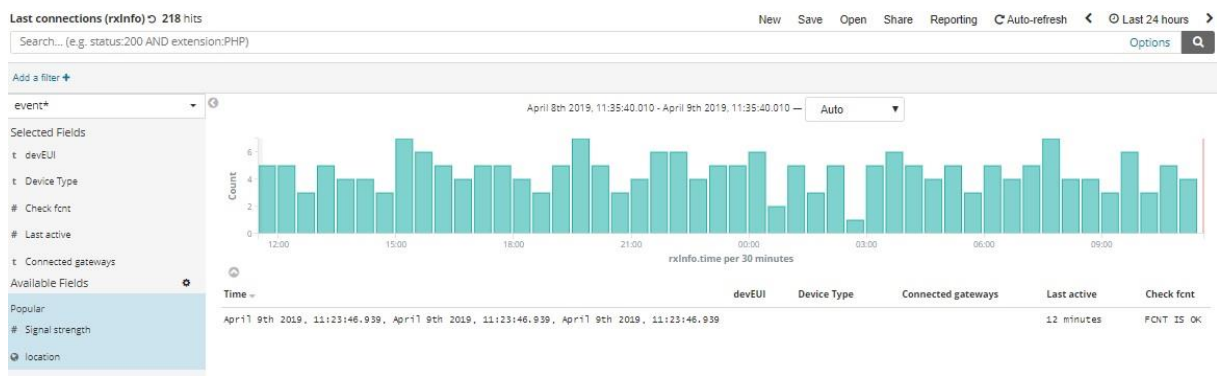
Joonis 29. Seadmete tüüp Pie Chart

Visuaal Vertical Bar (vt Joonis 30) kuvab kasutajale konkreetse seadme pakettide järjenumbreid ühenduse kvaliteedi kontrollimiseks. [19]



Joonis 30. Pakettide järjenumbri Vertical Bar

Visuaal saved search võimaldab ülevaatliselt viimati saabunud pakette. Kuvatakse timestamp, devEUI, Device Type, Connected gateways Last active ja Check fcnt. [22]



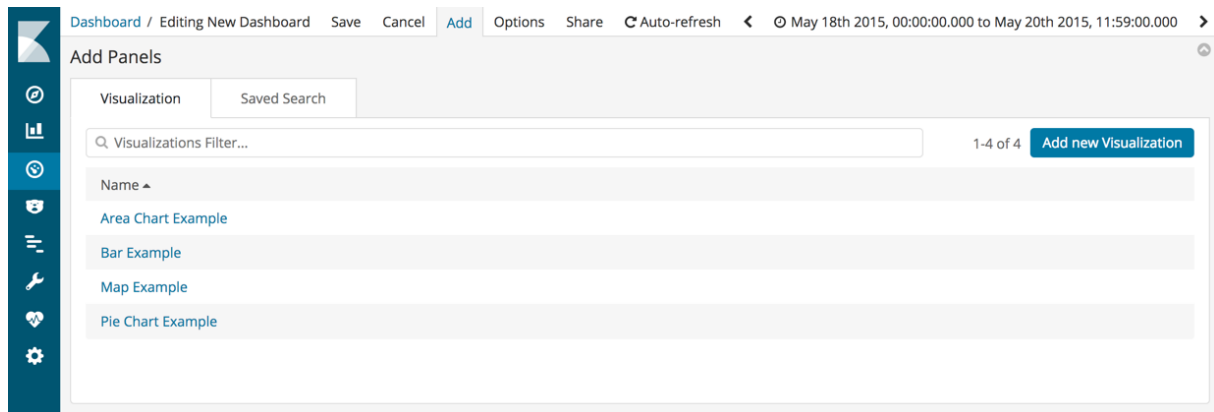
Joonis 31. Salvestatud otsingu kuvamine

### 3.9.3 Juhtpaneel loomine

Kibanas nimetatakse dashboard'ideks kogumit, kuhu kuuluvad eri visualisatsioonid ning salvestatud otsingud. Dashboard'id ehk juhtpaneelid koguvad loodud visuaalid üheks tervikuks. Visualisatsioonid võivad pärineda erivenatest andmeallikatest, kuid selles

juhtpaneelis on võimalik neist luua ühtne tervik. Igal juhtpaneelil on võimalik muuta korraga visualisatsioonide kuvatud ajavahemikku ja vajadusel ka mõnda muud filtrit.[23]

Joonisel 32 on kuvatud juhtpaneelide loomisega alustamist Kibanas, siin on võimalik lisada olemasoleva visualisatsiooni juhtpaneelile.

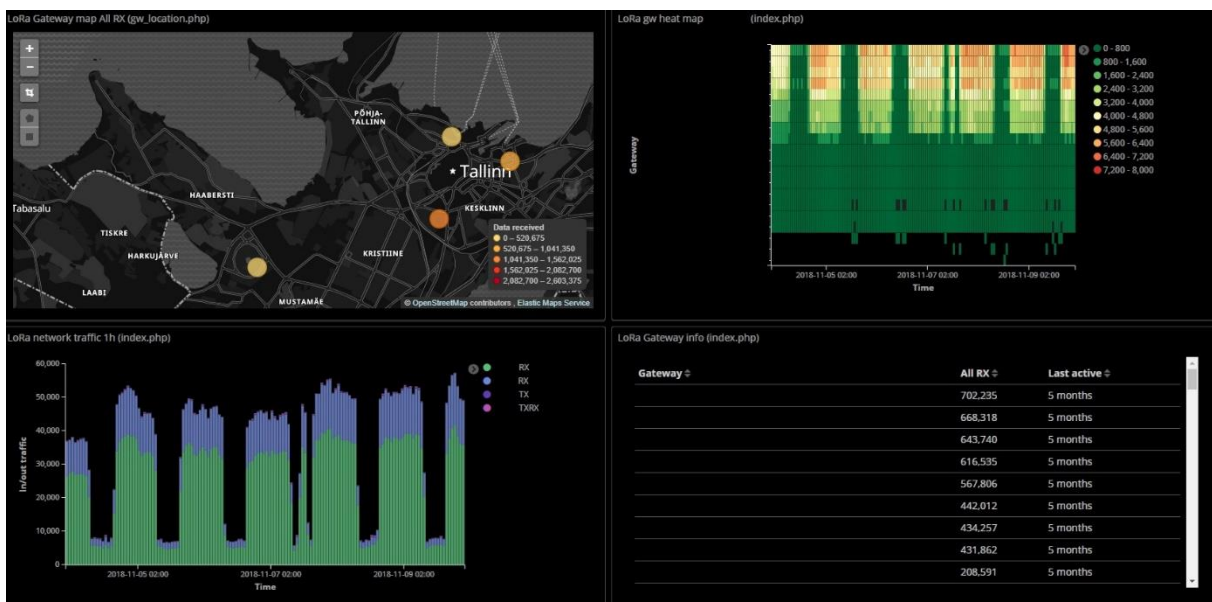


Joonis 322. Juhtpaneelide loomine Kibanas

## 4 Tulemused

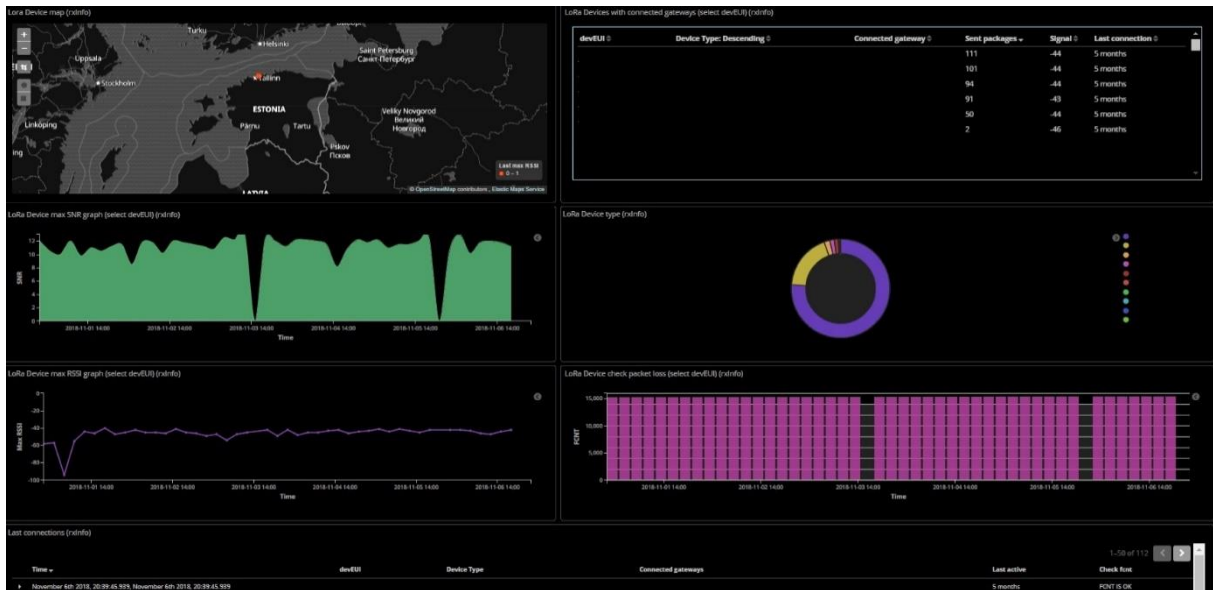
Töö tulemusi illustreerib kaks juhtpaneeli: üks neist puudutab tugijaamu ning teine on seotud võrgus olevate seadmetega.

Tugijaamade juhtpaneel (vt Joonis 33) võimaldab näha tugijaamade asukohta. Kollasest punaseni värvid näitavad tugijaamadele saabuvate andmete kogust, legendil on näha pakettide arvu vahemikku. Mida punasem on tugijaam seda hõivatum ta on. Joonisel kuvatud tabelil näeb tugijaamadelt saabunud pakettide arvu 24 tunni lõikes. Tabelist saab teada ka, et kui mõni tugijaam pole serverile infot edastanud. Väljakujunenud mustrite alusel on võimalik näha, millal võivad tugijaamad olla üle koormatud.



Joonis 333. Tugijaamade juhtpaneel

Teine juhtpaneel (vt Joonis 34) on seadmekeskne: kaart kuvab seadmete asukohta. Värvide punasest kollaseni näitavad signaalitugevust, kus punane on tugevam signaal ja kollane nõrgem signaal. Kui juhtpaneeli konkreetse seadme kaudu filtreerida, on võimalik graafikutel näha nende signaalinäitajaid ajas. Joonisel kuvatud tabelist saab vaadata, millisel seadmelt võivad esineda leviprobleemid, kui palju pakette on igalt seadmelt saabunud või millal toimus viimane ühendus seadme ja tugijaama vahel. Samuti saab jägida, kas paketid on saabunud järjest.



Joonis 344. Seadmete juhtpaneel

Loodud lahenduse kaudu on ettevõttel võimalik lihtsamalt andmeid analüüsida, jälgida tugijaamade ning seadmete asukohta kaardil; tuvastada mustreid tugijaamade koormatuse kohta ajas; kontrollida, kas andmed saabuvad järjest ning tuvastada, millistel seadmetel võib olla leviprobleeme.

## 5 Kokkuvõte

Lõputöö eesmärgiks oli luua olemasolevate andmete põhjal visualisatsioonid, millega ettevõtte saaks olemasolevaid andmeid analüüsida ning seeläbi enda teenust parendada. Seega oli vaja andmed kokku koguda ja visualiseerida, et tuvastada võrgus esinevaid probleeme.

Lõputöös kirjeldati nõuded platvormile, mille järgi otsustati, milline andmeanalüüsiplatvorm valida. Visualiseerimise võimalikeks alternatiivideks valikul olid Elastic Stack, Grafana, MS Power BI ja Thingsboard. Valiku nõuete osas oli oluline, see et analüüsiplatvorm oleks tasuta ning avatud lähtekoodiga. Seadmete ja tugijaamade asukoha monitoorimiseks oli vaja ka kaardirakenduse olemasolu. Samuti, kuna ettevõttes oli kasutusel Linuxi server siis oli vajalik, et süsteem töotaks sellel operatsioonisüsteemil. Oluline nõue andmeanalüüsiplatvormi valikul oli see, et ta omaks enda andmebaasi. Nende nõuete rahuldamise tõttu sai valituks Elastic Stack.

Töö käigus seadistati olemasolevate andmete jaoks platvorm ning visualiseeriti andmed. Töö lõpptulemuseks valmis kaks dashboard'i ehk juhtpaneeli, mille abil seoti erinevad visualisatsioonid ühtseks tervikuks.

Lõputöö oli orienteeritud andmete visualiseerimisele. Visuaalide loomisel keskenduti platvormile püstitatud nõuetele, mis tagas ka eesmärgi täitmise. Andmetest loodi visuaalid ning need seoti ühtsetesse juhtpaneelidesse. Loodud lahenduse kaudu on ettevõttel võimalik andmeid lihtsamalt analüüsida : jälgida tugijaamade ning seadmete asukohti kaardil; tuvastada mustreid tugijaamate koormatuse kohta ajas; kontrollida, kas andmed saabuvad järjest ning millistel seadmetel võib leviga probleeme olla. Tänu nõuete edukale täitmisele on nüüd ettevõttel kiirem ja kergem võimalus analüüsida IoT võrgus olevaid andmeid.



## Kasutatud kirjandus

- [1] M. Aparicio, Data Visualization, Instituto Universitario de Liboa, 2014. [WWW]  
[https://www.academia.edu/12294371/Data\\_Visualization](https://www.academia.edu/12294371/Data_Visualization) (22.03.2019)
- [2] A comparative study of LPWAN technologies for large-scale IoT deployment [WWW]  
<https://www.sciencedirect.com/science/article/pii/S2405959517302953>(22.03.2019)
- [3] A technical overview of LoRa® and LoRaWAN™ [WWW]  
<https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf>(22.03.2019)
- [4] ELK Stack Tutorial: Learn Elasticsearch, Logstash, and Kibana [WWW]  
<https://www.guru99.com/elk-stack-tutorial.html#3>(1.04.2019)
- [5] <https://www.loraserver.io/overview/>(1.04.2019)
- [6] Logstash Directory Layout [WWW] <https://www.elastic.co/guide/en/logstash/6.3/dir-layout.html>(10.04.2019)
- [7] How Logstash Works [WWW]  
<https://www.elastic.co/guide/en/logstash/6.3/pipeline.html>(10.04.2019)
- [8] Multiple Pipelines [WWW] <https://www.elastic.co/guide/en/logstash/6.3/multiple-pipelines.html>(10.04.2019)
- [9] Running Kibana on Docker [WWW]  
<https://www.elastic.co/guide/en/kibana/6.3/docker.html>(10.04.2019)
- [10] Elasticsearch, Logstash, Kibana (ELK) Docker image documentation [WWW]  
<https://elk-docker.readthedocs.io/>(10.04.2019)
- [11] Index Templates [WWW]  
<https://www.elastic.co/guide/en/elasticsearch/reference/6.3/indices-templates.html>(10.04.2019)
- [12] Geo-point datatype [WWW]  
<https://www.elastic.co/guide/en/elasticsearch/reference/6.3/geo-point.html>(10.04.2019)
- [13] Defining Your Index Patterns [WWW]  
<https://www.elastic.co/guide/en/kibana/6.3/tutorial-define-index.html>(10.04.2019)

- [14] Scripted Fields [WWW] <https://www.elastic.co/guide/en/kibana/6.3/scripted-fields.html>(10.04.2019)
- [15] Discover [WWW] <https://www.elastic.co/guide/en/kibana/6.3/discover.html>(10.04.2019)
- [16] Visualizing Your Data [WWW] <https://www.elastic.co/guide/en/kibana/6.3/tutorial-visualizing.html>(10.04.2019)
- [17] Coordinate Map [WWW] [https://www.elastic.co/guide/en/kibana/6.3/\\_coordinate\\_map.html](https://www.elastic.co/guide/en/kibana/6.3/_coordinate_map.html)(10.04.2019)
- [18] Heatmap Chart [WWW] <https://www.elastic.co/guide/en/kibana/6.3/heatmap-chart.html>(10.04.2019)
- [19] Line, Area, and Bar charts [WWW] <https://www.elastic.co/guide/en/kibana/6.3/xy-chart.html>(10.04.2019)
- [20] Data Table [WWW] <https://www.elastic.co/guide/en/kibana/6.3/data-table.html>(10.04.2019)
- [21] Pie Charts [WWW] <https://www.elastic.co/guide/en/kibana/6.3/pie-chart.html>(10.04.2019)
- [22] Saving and Opening Searches [WWW] <https://www.elastic.co/guide/en/kibana/6.3/save-open-search.html>(10.04.2019)
- [23] Building a Dashboard [WWW] <https://www.elastic.co/guide/en/kibana/6.3/dashboard-getting-started.html>(10.04.2019)

## Lisa 1 Mappingu template näide

```
PUT _template/template_3
{
  "index_patterns": ["status*"],
  "settings": {
    "number_of_shards": 5

  },
  "mappings": {
    "doc": {
      "properties": {
        "@timestamp": {
          "type": "date"
        },
        "@version": {
          "type": "text",
          "fields": {
            "keyword": {
              "type": "keyword",
              "ignore_above": 256
            }
          }
        }
      },
      "altitude": {
        "type": "long"
      },
      "channelConfigurationID": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
```

```
    "ignore_above": 256
  }
}
},
"createdAt": {
  "type": "date"
},
"description": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
},
"firstSeenAt": {
  "type": "date"
},
"gwRxPacketsReceived": {
  "type": "long"
},
"lastSeenAt": {
  "type": "date"
},
"location": {
  "type": "geo_point"
},
"mac": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
```

```
    "ignore_above": 256
  }
}
},
"name": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"organizationID": {
  "type": "text",
  "fields": {
    "keyword": {
      "type": "keyword",
      "ignore_above": 256
    }
  }
},
"updatedAt": {
  "type": "date"
}
}
}
}
```